INFOCLUB

REVISTA

DE INFORMATICA SI CALCULATOARE

1 1991



CUPRINS
CONTENTS

©INFOCLUB 1/91

4 **Flash:** Vreţi să cumpăraţi un laptop computer?

9 **Şah:** Şah-computer

13 **Anchetă:** Informatica în România

15 **Contribuţii:** Proiectul „Elect' 90"

16 **Actualitatea PC:** Elemente practice de grafică EGA/VGA

23 **Spot:** Placa Hercules II

28 **Computer world:** Borland umple ferestrele goale...; IMB lasă frîu liber...;

30 **Laborator Spectrum:** Rutină grafică pentru umplerea contururilor

32 **Ghidul utilizatorului:** MS-DOS pentru calculatoare personale; Turbo Pascal, versiunile 5.0 şi 5.5;

40 **Computer world:** NeXT din nou în... actualitate

43 **Atelier Spectrum:** Bright pentru HC-85

44 **Aplicaţii pentru toţi:** Un instrument de lucru în redacţia noastră — calculatorul!

Revistă trimestrială de informatică şi calculatoare

ANUL II — NUMĂRUL 2

ADRESA: Piaţa „Presa liberă" nr.1, 79781 Bucureşti

TELEFON: 17 72 44 sau 17 60 10, interior: 1151-1258

Comitetul Director

Ioan ALBESCU
Gheorghe BADEA
Mihaela GORODCOV

Colegiul Ştiinţific

Dr. mat. Stelian NICULESCU (Ministerul Învăţămîntului şi Ştiinţei); dr. ing. Nicolae ŢĂPUŞ şi dr. ing. Valeriu IORGA (Institutul Politehnic Bucureşti, Fac. de Automatică); cercet. ing. Eugen GEORGESCU şi cercet. Ion DIAMANDI (Institutul de Tehnică de Calcul).

Culegere pe calculator: Elena Vasilief; Mariana Badea

Corectură: Mariana Badea

Prezentare grafică: Gabi Cătălinoiu

Administraţia: Editura „Presa Liberă"

Tiparul: Întreprinderea „Arta Grafică"

Abonamentele se pot efectua pe adresa redacţiei prin mandat poştal pe numele Badea Gheorghe, urmînd ca abonamentul sa fie expediat prin poşta la adresa indicata.

Pentru cititorii din instituţii, unitaţi de invaţamint, intreprinderi de stat şi particulare numarul minim de abonamente este de 50 exemplare/apariţie, putind beneficia, in acest fel, de o reducere de 20%. Expedierea „abonamentului colectiv" se va face de catre redacţie prin colet poştal la adresa indicata.

Preţ de vînzare: 35 lei

IDG
INTERNATIONAL DATA GROUP

Avind sediul in Boston, Massachusetts, INTERNATIONAL DATA GROUP este liderul mondial cu privire la serviciile informaţionale şi la tehnologia obţinerii informaţiei, cu un venit anual de 620 milioane US $ şi 3 800 de angajaţi.

Divizia dedicată expoziţiilor, WORLD EXPO CORPORATION organizează 40 de expoziţii şi conferinţe de calculatoare în 18 ţări.

Divizia sa de publicistică şi editare, IDG COMMUNICATION publică 150 de ziare şi reviste în 50 de ţări. Divizia de cercetare, INTERNATOINAL DATA CORPORATION (IDC) este liderul mondial al analizei şi prospectării de marketing în domeniul calculatoarelor.

INFOCLUB este o publicaţie a International Data Group (IDG), cel mai mare editor de reviste de informatică şi calculatoare din lume. În fiecare lună, 25 de milioane de oameni citesc una sau mai multe publicaţii IDG.

Publicaţiile IDG includ: ARGENTINA: Computerworld Argentina; ASIA: Computerworld Hong Kong, Computerworld Southeast Asia, Computerworld Malaysia, Computerworld Singapore, Infoworld Hong Kong, Infoworld SE Asia; AUSTRALIA: Computerworld Australia, PC World, Macworld, Lotus, Publish; AUSTRIA: Computerwelt Oesterreich; BRASILIA: DataNews, PC Mundo, Automacao and Industria; BULGARIA: Computerworld Bulgaria, Computer Magazine; CANADA: ComputerData, Direct

ECOURI ŞI... CERTITUDINI

Aşadar, iată-ne din nou împreună, la cel de-al doilea număr al revistei INFOCLUB!
Am primit la redacţie sute de telefoane şi scrisori cu sugestii, observaţii, critici şi chiar... laude şi încurajări pe care nu am vrut să le punem în capul listei pentru a nu părea lipsiţi de modestie.
Am acordat prea puţin spaţiu calculatoarelor Sinclair Spectrum? Se poate! Şi putem invoca aici scuza spaţiului prea restrîns şi a faptului că am dorit să acoperim cît mai multe tipuri de calculatoare.
Am tratat insuficient problema PC-urilor? Perfect adevărat, în mare parte din aceleaşi motive!
Unii ne-au reproşat chiar spaţiul prea restrîns dedicat noutăţilor din domeniu! Dar, INFOCLUB-ul doreşte să fie o revistă orientată spre aplicaţii practice, lăsînd noutăţile şi articolele generale pe seama... altor publicaţii.
De aceea, noi, am încercat şi vom încerca să acoperim cît mai mult, atît prin intermediul INFOCLUB-ului, cît şi al suplimentelor sale dedicate, astfel, încît să reuşim să satisfacem un cît mai mare număr de cititori şi utilizatori de tehnică de calcul.
Avem posibilitatea să o facem şi, chiar, certitudinea că o putem realiza! Şi iată de ce!
După cum cred că aţi aflat din diferite surse mass-media, INFOCLUB este membru IDG, International Data Group - USA, cel mai important editor de reviste de informatică din lume.
Binecunoscut pe întreg mapamondul, IDG este sinonim, pentru milioane de cititori şi utilizatori, cu titluri binecunoscute de publicităţii începînd cu Computer World, fondată în 1967, cu Info World, PcWorld, MacWorld, Network World, Digital News, Federal Computer Week şi multe multe altele şi terminînd cu cele mai recent înfiinţate, mai ales în Europa răsăriteană.
Reviste, ziare, publicaţii de informaţie în domeniu, de la jocuri pînă la aplicaţii de ultimă oră apar sub antetul IDG în America Latină, Australia, Noua Zeelandă, China, Japonia, Hong Kong, Europa, Statele Unite ale Americii etc. (aşa cum reiese şi din caseta tehnică).
Informaţia privită ca o importantă resursă economică şi educarea „la zi" a oamenilor cu tot ce se întîmplă într-un domeniu atît de dinamic şi de vast cum este informatica, sînt cîteva dintre principiile de bază ale IDG.
Revista INFOCLUB, în calitate de membru al IDG se bucură de numeroase avantaje!
Vehicularea unei documentaţii bogate şi variate din care, începînd cu acest număr, am selectat pentru dumneavoastră cîteva fragmente referitoare la noua linie NeXT şi la unele noutăţi grupate sub genericul, binecunoscut deja - „Computerworld".
De asemenea, posibilitatea de a transmite către IDG aşa numitele „contribuţii" şi realizări deosebite ale specialiştilor noştri în domeniul, fireşte, al calculatoarelor. De aceea, folosim încă odată acest prilej pentru a invita utilizatorii din ţară să ne remită pe adresa redacţiei astfel de materiale spre a fi transmise în reţeaua de noutăţi a IDG. Este o pîrghie importantă cu multiple avantaje şi pe care vă invităm să o folosiţi, o deschidere ce poate fi benefică pentru ce s-a realizat şi se va realiza în continuare în acest domeniu în ţara noastră.
Despre IDG se pot scrie foarte multe! O companie „tentaculară", cu trăsături aparte în peisajul general al marilor grupuri, care pune un accent deosebit pe educarea şi şcolarizarea personalului.
„Cred că noi investim în instruirea personalului mai mult decît alte companii. Este important pentru succesul nostru. Am ajuns la concluzia că cele mai bune rezultate ale unei investiţii vin din şcolarizare şi educaţie", *a spus, într-un interviu, preşedintele şi fondatorul IDG,* **domnul Patrick McGovern.**
Aceasta, alături de contactele şi colaborările companiei cu firme de primă mărime şi importanţă în tehnica de calcul mondială, constituie cheia succesului IDG.
În ceea ce ne priveşte, INFOCLUB rămîne o revistă deschisă tuturor sugestiilor şi dialogului cu utilizatorii şi cititorii care încearcă să se adapteze în funcţie de dinamica domeniului.
Cam atît deocamdată. Şi vă aşteptăm în continuare!

Access, Graduate CW, Macworld; CHILE: Informatica, Computacion Personal; COLUMBIA: Computerworld Columbia; CEHOSLOVACIA: Computerworld Cehoslovacia, PC World; DANEMARCA: CAD/CAM WORLD, Computerworld Danmark, Communication World, PC World, Macworld, Unix World, PC LAN World; FINLANDA: Mikro PC, Tietoviikko, Tietotekniika; FRANŢA: Le Monde Informatique, Distributique, InfoPC, Telecoms International; GERMANIA: Computerwoche, Information Management, Amigawelt, PC Woche, PC Welt, Unix Welt, Macwelt RD; GRECIA: Computerworld, PC World, Macworld, Infoworld; UNGARIA: Computerworld SZT, Mikrovilag; INDIA: Computers and Communications; ISRAEL: People and Computers; ITALIA: Computerworld Italia, PC World Italia; JAPONIA: Computerworld Japan, Macworld; COREEA: Computerworld, PC World; MEXIC: Computerworld Mexico, PC Journal; OLANDA: Computerworld Netherland, PC World, Amiga World; NOUA ZEELANDĂ: Computerworld New Zealand, PC World New Zealand; NIGERIA: PC World Africa; NORVEGIA: Computerworld Norge, PC World Norge CAD/CAM, Macworld Norge; CHINA: China Computerworld, China Computerworld Monthly; FILIPINE: Computerworld Phillippines, PC Digest/PC World; POLONIA: Komputers Magazine, Computerworld; **ROMÂNIA: Infoclub;** SPANIA: CIM World, Communicaciones World, Computerworld Espana, PC World, Amiga World; SUEDIA: ComputerSweden, PC/Nyheterna, Mikrodatorn, PC World, Macworld; ELVEŢIA: Computerworld Schwitzerland, Macworld; TAIWAN: Computerworld Taiwan, PC World, Publish; TAILANDA: Computerworld; TURCIA: Computerworld Monitor, PC World/Turkiye; MAREA BRITANIE: Graduate Computerworld, PC Business World, ICL Today, Lotus UK, Macworld U.K.; STATELE UNITE: Amiga World, A+, CIO, Computerworld, Digital News, Federal Computer Week, GamePro, IDG Books, InfoWorld, Macworld, NextWorld, Network World, PC Games, PC World, Portable Office, PC Letter, Publish, Run, Sun Tech Journal; URSS: MIR PC, Computerworld URSS, Network, Manager Magazine; VENEZUELA: Computerworld Venezuela, Micro Computerworld; YUGOSLAVIA: Moj Mikro.

laptop

COMPUTER

micro



Cristian IONESCU

VREȚI SĂ CUMPĂRAȚI UN LAPTOP-COMPUTER? FOARTE BINE, DAR...

Utilizarea în masă a calculatoarelor personale compatibile modelelor XT sau AT a implicat pentru producători un foarte incitant pariu: realizarea unui PC care să poată fi luat în călătorie și să servească drept bază mobilă de calcul. Acest computer trebuia să poată fi utilizat practic oriunde, așezat pe cele mai diferite suprafețe, cel mai adesea ținut chiar... pe genunchi. Un computer complet, ținut și operat pe genunchi! Numele se născuse: laptop computer. Mai trebuia să fie și produs!



Prima problemă care trebuia rezolvată era aceea a contradicțiilor care există între diferitele module constructive, deja standardizate în tehnica de calcul. Procesoarele din seria 8086, 80286 și 80386, floppy discurile de 3,5 inch, discuri hard, afișaje contrastante, dar economice în consum, tastatura cît mai ergonomică și insensibilă la contaminanți, toate acestea trebuiau incluse într-o carcasă care să ofere bune posibilități de răcire, să asigure ecranarea radiației electromagnetice, precum și stabilitate structurală. În carcasă mai trebuia făcut loc și unei surse de alimentare care să fie simultan rezistentă climatic, ușoară, etanșă, să aibă o viață cît mai lungă, să asigure densități de energie mari pe volum și să poată fi încărcată rapid.

Tot acest ansamblu trebuie să reziste la proba uzuală de cădere de la înălțimea unui birou pe o pardoseală de ciment, indiferent de partea care vine prima în contact cu solul. În final, produsul trebuie să asigure compatibilitate software cu modelele de PC compatibile IBM.

Simpla înșiruire a cerințelor care au stat în fața proiectanților indică obstacolele formidabile care au trebuit depășite pentru realizarea acestor bijuterii tehnice. Și aici, ca în oricare alt domeniu, în care ingeniozitatea miniaturizării a depășit limitele aparente al posibilului, primii producători industriali de computere laptop au fost japonezii cu legendara serie **Toshiba T 1000, T 1200,** etc. Această serie de succes comercial a evoluat ajungînd astăzi pînă la modelele **T 5100** și **T 5200** cu discuri hard de 100 și 200 MBytes.

UN LAPTOP ÎNSEAMNĂ...

Procesoare. Cu predilecție sînt folosite procesoarele V20 (compatibil 8086) sau cele realizate în tehnologia CMOS: 80C86 și 80C286. Cu toate acestea se întîlnesc și modele „sport" care demonstrează solida performanță a unui 386SX cu 16 MBytes de memorie operativă.

Afișajul. Computerele de tip laptop sînt obligate să folosească tipuri de afișaje caracterizate prin consumuri reduse: cristale lichide, plasmă sau chiar tuburi plate color, avînd tunul electronic așezat lateral, așa cum este cazul modelului Hitachi HL 500C, unde rezoluția este 640 x 480 cu 8 culori!

Standardele implementate sînt, în general, orientate în două direcții principale. În primul rînd, pentru afișaje ieftine cu cristale lichide se folosesc formate alfanumerice și grafice nepretențioase, care nu respectă nici un sandard grafic cunoscut. Pentru mode-

lele mai evoluate se folosesc afişaje compatibile EGA şi VGA monocrom şi, rar, color.

Pentru asigurarea consumurilor reduse, ecranele sînt stinse după o perioadă de inactivitate a operatorului; prima apăsare pe taste va readuce conţinutul ecranului înapoi.

Unităţile de floppy-disc. În calculatoarele de tip laptop s-a standardizat existenţa unui floppy de 3,5 inch, cu o capacitate de 1,44 MBytes. În tipurile cu consum ultra-redus s-au introdus RAM-Cards, plăci pe care sînt montate memorii RAM de 32, 64 sau 128 KBytes şi care pot fi conectate la computer prin intermediul unui conector adecvat. Aceste RAM-Cards, deşi rezolvă parţial nevoia unei memorii mai mari, au o serie de dezavantaje: nu toate pot reţine datele şi nu sînt standaridizate.

Unităţile cunoscute de 5,25 inch, pot fi folosite ca echipamente externe cu sursă proprie de alimentare şi cu cabluri de conectare externe. Fără îndoială, acest tip de utilizare reduce mult caracteristica de portabilitate, dar asigură compatibilitatea cu suportul de informaţii, cerinţă primordială în multe aplicaţii.

Unităţile de hard-discuri. Pe lîngă cunocutele unităţi e 3,5 inch, în laptops au apărut şi unităţi de 2,5 inch. Capacităţile oferite nu sînt deloc modeste, începînd cu 40 MBytes pînă la discuri „serioase" de 100 MBytes care pot memora între 20 000 şi 50 000 pagini A4 dactilografiate. Volumul şi greutatea acestor pagini memorate depăşesc cu mult pe cele ale computerului. Totodată, pentru aplicaţii pretenţioase sînt livrabile şi unităţi externe de discuri de 100 MBytes, cu surse proprii de alimentare.

Tot ca unităţi externe sînt livrabile şi unităţi de casetă magnetică tip streamer, extrem de utile în salvări sau arhivări ale conţinutului unor hard discuri.

Memoria internă. Dacă nimic nu ne mai surprinde în tehnologia actuală, atunci capacitatea memoriilor interne apare ca un argument uzual: 1 MByte, extensibil la 4 sau chiar la 16 MBytes! Programele cele mai exigente şi „infometatе" de memorie încap acum în „colegul" nostru electronic cu ajutorul căruia, în drumul de la Bucureşti la Braşov — atît durează acumulatorii — putem să proiectăm, simulăm, corectăm şi trasăm placa noului circuit electronic pe care îl oferim spre integrare în noul şi ultramodernul camion...

Comunicaţii. Modelele laptop îşi dovedesc utilitatea mai ales în cazurile în care posesorul are o activitate care reclamă mobilitate. În mod normal, în fiecare laptop întîlnim o interfaţă serială, multe dintre ele posedînd şi un modem încorporat care lucrează la vitezele uzual întîlnite, deci pînă la 2400 Bauds. La fel de des întîlnit este şi cazul existenţei unei cuple V.24 la care se poate ataşa un modem acustic portabil livrat cu tot cu cupelele de izolare acustică care se montează pe receptorul telefonic. Mai nou, producătorii oferă şi plăci de tip telefax la un pachet soft adecvat.

Un exemplu este Toshiba Microlink 2 400XL care comunică destul de modern: V.21 duplex la 300 bit/s, V.22 cu 1 200 şi 2 400 bit/s duplex. În plus, placa mai lucrează şi în mod Fax folosind CCITT V.27 semiduplex, la viteze de 2 400 sau 4 800 bit/s.

Oricum ar fi, placa de comunicaţii, modemul precum şi pachetul soft trebuie alese după o consultare cu poşta naţională pentru a evita orice surprize. Trebuie remarcat că nu orice laptop sud-est-asiatic „ieftin" este capabil să se înţeleagă fără probleme cu un Felix-PC autohton sau cu importuri „standard PC". Situaţia se agravează dacă vrem să comunicăm cu maşini de la nivelul unui minicalculator în sus.

Aceeaşi situaţie se întîlneşte şi în cazul aplicaţiilor de tip Fax. Şi aici, regula de aur: „întreabă de o mie de ori şi cumpără o dată!" se dovedeşte a fi deosebit de utilă. Mai ales că un laptop nu este tocmai ieftin...

Greutatea, dimensiunile şi, mai ales, costul! În general, computerele laptop nu sînt tocmai uşoare. Trebuie să ne aşteptăm la greutăţi cuprinse între 3 şi 6 kg, în funcţie, mai ales, de tipul acumulatoarelor folosite.

Costurile sînt în general mai ridicate decît ale calculatoarelor personale de birou. De exemplu, un PC uzual, tip AT, poate costa pe piaţa germană în jur de 2 500 DM, dar orice încercare de apropiere de un laptop model AT ne va face să numărăm la casă peste 5 000 DM! Fără îndoială şi aici performanţa se plăteşte. Şi încă scump... Un calcul simplu arată că 1 kg laptop costa între 800 şi 2 000 DM!

Sistemele de alimentare. Pentru a putea asigura alimentarea unui laptop trebuie să se facă o alegere deosebit de dificilă, mai ales din punctul de vedere al producătorului. Fără îndoială, produsul trebuie să fie: rapid, cu memorie operativă mare, cu ecran vizibil şi în lumină directă, cu memorie mare pe discuri, dar, totodată, uşor, independent de surse de energie, insensibil la temperatură, umiditate, şocuri.

Toate aceste cerinţe primare creează un tablou contradictoriu, rareori rezolvat satisfăcător. Vom întîlni astfel modele laptop cu consum redus, la care bateria susţine sistemul peste 60 ore, dar cu procesoare arhaice — Z80 — sau modele ultraputernice ale căror acumulatoare ajung doar pentru circa 2 ore. Totodată, tipul acumulatoarelor variază foarte mult, de la grelele baterii cu plumb-acid pînă la periculoasele baterii cu litiu. Practic, deşi un laptop este portabil, mai bine folosim adaptorul şi îl alimentăm la reţea!

Acest tablou relativ sumbru trebuie totuşi completat cu faptul că utilizatorul trebuie să acorde o atenţie deosebită acumulatoarelor: o scurgere minoră de electrolit poate distruge iremediabil computerul. La multe modele, o funcţie software asigură indicarea gradului de încărcare a acumulatoarelor, iar sub o anu-

mită limită, computerul avertizează sonor și vizual necesitatea încărcării acestora înainte ca să fie distruși prin descărcare excesivă.

Software. Dacă PC-laptop pe care îl achiziționați conține un procesor Intel care depășește nivelul 8086 precum și un hard disc, atunci veți avea acces la sisteme de operare evoluate: DOS 3.xx sau DOS 4.01. Prin intermediul acestora puteți asigura rularea unui tezaur de pachete soft care acoperă, practic, toate aplicațiile uzuale. Fără îndoială, trebuie verificată utilizarea oricărui pachet înainte de generalizare în sistemul companiei în care lucrați.

A doua posibilitate constă în a avea un laptop care are un sistem de operare propriu. În acest caz este uzual să fie incluse un procesor de texte, o bază de date, un program de calcul cu tabele tip Lotus, un interpretor Basic și un modul de comunicații. Pe lîngă acestea, mai pot fi întîlnite și pachete de grafică comercială sau de gestiune a unui fișier de adrese, cu posibilitatea de a folosi modemul încorporat direct pentru formarea numărului de telefon. În acest al doilea caz, este uzuală funcția de calculator științific și comercial analog modelelor produse fie de Hewlett-Packard, fie de Texas Instruments.

În ambele variante, ținînd cont de caracterul de portabilitate al mașinii, puteți să obțineți și pachete care oferă o interfață comodă între utilizator și sistemul de operare instalat. Asemenea module, numite „Shell" sau „Personal application manager" consumă spațiu în memoria internă și pe discul hard dar scutesc utilizatorul de manevre de prestidigitator pe o tastatură densă și cu asignări multiple.

SURPRIZĂ SAU NOUA VIAȚĂ A UNUI VETERAN.

Aparent, în lumea PC, nu mai există procesoare decît de la 80286 în sus. Totuși, firma Rhisc — bun exemplu și pentru producătorii noștri — oferă un laptop de dimensiunile unui pagini A4, cîntărind sub 2 kg, are 5 cm înălțime, bateriile îl susțint 60 ore, RAM-card CMOS și display LCD de 8 linii și 80 caractere.

Ce procesor este folosit? Veteranul Z80 care, la 3 MHz s-a dovedit capabil să rezolve la preț redus — tot computerul costa sub 1 000 DM, în varianta de bază — toate problemele legate de gestionarea perifieriei și de pachetele software integrate în produs.

Un exemplu recent.

Un exemplu de laptop tipic îl constituie **Dell 316LT**, un laptop care folosește un procesor 386SX, la 16 MHz, cu o memorie internă de I MByte — modelul de bază — un hard-disc de 40 MBytes și un floppy de 3,5 inch. Afișajul este asigurat cu cristale lichide supertwist și asigură confortul standardului VGA: 640 x 480 puncte. În model mai găsim și o interfață serială și una paralelă, precum și un slot de 8 biti pentru extensii tip XT, cum ar fi: rețele locale, fax sau modem. Plătind peste 8 000 DM, puteți cumpăra acesth„compumonster" cîntărind 7,5 kg și cu dimensiuunile 33 x 36 x 8,5 cm.

Mic îndrumar de prețuri.

Fără îndoială, estimarea utilității unui produs se face considerînd și prețul pe care îl plătim pentru serviciile sale. Pentru aceasta v-am putea oferi un tabel de prețuri practicate pe piața europeană cea mai apropiată de noi(!): piața germană, anul 1990. Prețurile fiind exprimate în DM, iar caracterul lor pur informativ, pot exista și prețuri mai mari sau chiar mai mici, mai ales în cazul produselor zise de „export".

Cum sub acest eufemism se ascund produse care nu au trecut normele de atestare tehnică pentru piața respectivă, amînăm intenția noastră pentru altă dată, sau vă putem satisface curiozitatea numai ca urmare a solicitărilor dumneavoastră.

Computerworld

● Poliția din Nevada și-a îmbunătățit de curînd rețeaua de telecomunicații care permite ofițerilor și patrulelor de pe stradă să se conecteze la bazele de date locale, federale și de stat în circa 10 secunde! Cu o singură cerere ofițerii de pe teren, aflați în orice punct al statului pot avea acces foarte rapid la un sistem specializat (LEMS = Lan Enforcement Message Switching system) la informațiile cele mai diverse privind infractori, criminali, infracțiuni etc., pentru elucidarea în timp foarte scurt a cazurilor și anchetelor. Înainte de implementarea acestui sistem, utilizatorii trebuiau să acceseze 17 baze de date din diferite puncte ale statului pentru aflarea informațiilor dorite.

LEMS ia în considerare centrele de informație și le „îndrumă" la sursele de date adecvate mărind mult viteza întregului proces precum și volumul tranzacțiilor (circa 200 000 pe zi!) care se pot face într-o unitate dată de timp, deoarece „viteza și flexibilitatea sînt esențiale pentru prinderea unui criminal". (Network World — 31 decembrie 1990/7 ianuarie 1991).

● În noiembrie trecut Motorola a anunțat lansarea și producerea puternicului microprocesor 68040, despre care Apple a afirmat că va fi „inima" viitorului Macintosh. Avînd ceasul de 25 MHz, microprocesorul echipează deja și stațiile de lucru Hewlett-Packard și NeXT, asigurîndu-le o viteză de lucru de 20 MIPS; M 68040 este de cîteva ori mai rapid decît predecesorul său M 68030 de 40 MHz care echipează Macintosh II fx. (Macworld — februarie 1991).

Computerworld


şah
computer
KINGDOM

Viorel DARIE

ŞAH — COMPUTER

Vă mai amintiţi de ASTRO—64? Vă mai amintiţi de acele pasionante articole realizate de U. Vălureanu, în paginile revistei „Magazin", cu privire la evoluţia programelor de şah pe calculator? În acei ani şahul pe calculator îşi făcea debutul în ţara noastră, eram printre puţinele ţări care aveau realizate asemenea programe. În ultimii ani, însă, din motive mai mult sau mai puţin obiective, şahul programat din ţara noastră a cunoscut o vizibilă stagnare, în timp ce, în mai multe ţări din lume această disciplină a informaticii s-a dezvoltat mult, ducînd la apariţia de numeroase programe de şah, care participă la campionate naţionale şi la frecvente întreceri internaţionale.

APEL CĂTRE INFORMATICIENI

Prin acest material, care va apărea în mai multe numere ale revistei „INFOCLUB", intenţionăm să iniţiem o interesantă acţiune, ca toţi cei care se simt atraşi de subiectul şahului programat (informaticieni, studenţi, elevi) care au posibilitatea să lucreze pe un calculator personal, să încerce să-şi realizeze propriile programe de şah.

În această idee, vom prezenta un mini—curs de programare a şahului, sub forma mai multor articole ce vor apărea în această revistă. Vă rugăm să popularizaţi iniţiativa noastră printre prietenii voştri informaticieni!

Cursul nostru, deşi teoretic, sperăm să constituie un bun ghid pentru înţelegerea problematicii şahului programat, care ar putea să-i facă pe începătorii în acest domeniu să evite greşelile de concepţie, care ar reduce, din start, performanţele programelor lor. De bună seamă, fiecare autor e liber să-şi aleagă propriile căi de abordare a acestui inepuizabil subiect, cursul nostru însemnînd, de fapt, o colecţie de sugestii.

Daca în urma acestei iniţiative vor fi scrise mai multe programe de şah, intenţionăm să organizăm cu ele campionate locale sau naţionale. De asemenea, cele mai bune dintre aceste programe vor fi desemnate să participe la campionate (sau meciuri) internaţionale.

Curaj şi succes tuturor acelora care se decid să abordeze acest frumos subiect, al programelor de şah pe calculator! Puteţi să ne scrieţi pe adresa revistei cu menţiunea „ŞAH—COMPUTER".

Sfaturi de început.

● Dacă lucraţi pe microcalculatoarele pe 8 biţi (CUB—Z, M—118, SINCLAIR etc.) se pare că nu poate fi evitat limbajul de programare ASSEMBLER, dacă se urmăreşte performanţa de timp, căci, chiar şi limbajele C şi PASCAL pe aceste microcalculatoare generează cod ineficient. Eventual, se poate scrie în aceste limbaje cu scopul de a depana programele, ca apoi ele să fie trecute pe calculatoare mai puternice.

● Pe microcalculatoarele pe 16 sau 32 de biţi însă (cum sînt cele compatibile IBM—PC), compilatoarele C şi PASCAL generează un cod deosebit de eficient şi nu se mai justifică nici o reticenţă în privinţa eficienţei acestor limbaje. E drept, programînd în ASSEMBLER s-ar putea să se mai optimizeze cîte ceva, nesemnificativ totuşi. Este o iluzie că programarea în limbajul de ansamblare ar aduce avantaje, de fapt, asta face să pierdeţi mult timp cu implementarea oricărei idei, şi veţi avea de depanat multe erori.

● În plus, programînd în limbajele C sau PASCAL, veţi putea beneficia de posibilitatea apelului recursiv al procedurilor, lucru neoferit de FORTAN, COBOL sau ASSEMBLER.

● Evitaţi, pe cît posibil, folosirea numerelor reale într-un program de şah, căci operaţiile cu numere reale consumă mult timp. De fapt, numerele reale pot fi evitate aproape integral într-un program de şah.

● Pentru codificarea tablei de şah, a numerelor pătratelor, a indicatorilor, puteţi folosi variabile pe un octet. Totuşi, pentru exprimarea cît mai fină a tuturor valorilor oferite de diferite criterii de apreciere a **ponderilor statice** şi a **valorilor mini-max**, e bine ca aceste numere să fie întregi, pe 16 biţi.

MODEL INFORMATIC AL JOCULUI DE ŞAH

Aşadar, cu ce începem scrierea unui program de şah?

De bună seamă, primul lucru ce se cere lămurit, este modul de reprezentare în memoria calculatorului a tablei de şah şi a poziţiilor pieselor pe tabla de şah.

Alegerea modelului pentru tabla de şah.

Pare banala aceasta problemă, căci ce poate fi mai simplu decît o matrice TABLA (8,8), ale cărei elemente sînt numere întregi, ce codifică diferite tipuri de piese?

Totuşi, cînd scriem un program de şah, tabla în forma ei cea mai simplă de matrice 8 x 8 are inconvenientul că, în mutarea succesivă a unei piese dintr-un pătrat în altul, trebuie făcute 4 teste de fiecare dată, dacă acea piesă n-a depăşit marginea tablei, în sus, în jos, în stînga şi în dreapta. În plus, accesul la orice element al unei matrice cu două dimensiuni TABLA (i,j) ascunde şi o înmulţire i x 8 + j.

	6E	6F	70	71	72	73	74	75	76	77
	64	65	66	67	68	69	6A	6B	6C	6D
8	5A	5B	5C	5D	5E	5F	60	61	62	63
7	50	51	52	53	54	55	56	57	58	59
6	46	47	48	49	4A	4B	4C	4D	4E	4F
5	3C	3D	3E	3F	40	41	42	43	44	45
4	32	33	34	35	36	37	38	39	3A	3B
3	28	29	2A	2B	2C	2D	2E	2F	30	31
2	1E	1F	20	21	22	23	24	25	26	27
1	14	15	16	17	18	19	1A	1B	1C	1D
	A	B	C	D	E	F	10	11	12	13
	0	1	2	3	4	5	6	7	8	9
		A	B	C	D	E	F	G	H	

fig.1 Tabla cilindrică. Numerotarea în (hexa) a pătrătelor ei. Tabla de şah propriu-zisă este în chenarul interior.

Şah–computer

Acest motiv face ca să se recurgă la reprezentări ale tablei de şah mai convenabile, care să rezolve anumite probleme. Astfel, cel mai des se foloseşte aşa-zisa **tabla cilindrică,** care se poate vedea în figura I, care este un vector TABLA (120) conţinînd 120 de elemente, ce se consideră împărţită pe 12 linii a cîte 10 elemente, şi care conţine liniile şi coloanele tablei de şah 8 x 8, mărginite sus şi jos cu fişii a cîte două linii cu valori negative, iar în stînga şi dreapta cu încă o fişie cu valori negative, folosite pentru detecţia marginii. Dacă se îndoaie marginea din stînga a tablei, ca să se unească cu marginea din dreapta, se obţine o fişie cu două rînduri de pătrate cu valori negative, precum şi aspectul de tablă cilindrică, de unde şi denumirea. Fişiile de 2 rînduri din margine sînt necesare din cauza mutărilor calului.

Fişiile de margine conţin coduri negative (mod de detecţie a marginii tablei), iar celelalte elemente ale vectorului TABLA (120) conţin fie valori zero (pătrate libere), fie valori strict pozitive (codurile pieselor albe sau negre).

	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
8	-1		11							-2
7	-1				7			1	7	-1
6	-1						1		6	-1
5	-1	7			4				3	-1
4	-1	1		9		2		7		-1
3	-1	7		1	12		7	1		-1
2	-1	3			7		2			-1
1	-1		9		5	4				-1
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
		A	B	C	D	E	F	G	H	

fig. 2 Codificarea unei poziţii de pe tabla de şah. Mat în 2 mutări.

Codificara poziţiilor pe tabla e şah.

Ce coduri se atribuie diferitelor tipuri de piese?

Nu are prea mare importanţă acest lucru, totuşi, fiecare autor de programe de şah îşi alege astfel codificarea pieselor, încît să-şi rezolve anumite probleme curente: folosirea codului piesei ca index în diferite tabele, aprecierea importanţei piesei conform codului ei etc.

În programul ASTRO—64 (realizat pe calculatorul FELIX C—512), de asemenea în LABIRINT—64 şi ATOM—64 (realizate pe microcalculatoarele M—118, CUB—Z), s-a ales următoarea codificare a pieselor:

1 = pion alb	**7 = pion negru**
2 = turn alb	**8 = turn negru**
3 = cal alb	**9 = cal negru**
4 = nebun alb	**10 = nebun negru**
5 = regina albă	**11 = regina de negru**
6 = rege alb	**12 = regele negru**

Încercaţi să reconstituiţi poziţia de pe tabla de şah codificată în figura 2. În această figură se vede un cod „—2" după pătratul H8. Este o mică „invenţie", care permite detectarea sfîrşitului tablei de şah cînd aceasta este parcursă secvenţial.

Codificarea mutărilor.

O mutare este o înregistrare cu mai multe cîmpuri: pătratul sursă, pătratul destinaţie, indicator de mutare specială, ponderea (nota) asociată mutării etc.

Pătratele se codifică ca un singur întreg, pe un octet (sau pe doi octeţi pe calculatoare pe 16 biţi), conform numerotării din figura 1, de exemplu.

Generatorul de mutări.

Pentru a putea scrie un subprogram care să genereze lista tuturor mutărilor ce se pot efectua, într-o semimutare, dintr-o poziţie dată pe tabla de şah, este necesar să fie precizate următoarele informaţii:

— conţinutul vectorului TABLA (120);

— indicatorul care să spună dacă albul sau negrul este la mutare;

— alţi 6 indicatori care să precizeze dacă regii sau turnurile au mutat;

— ultima mutare a adversarului (pentru mutări „en-passant").

Mutările se obţin prin adăugarea la poziţia curentă a unei piese, a unui **increment** (pozitiv sau negativ), increment ce corespunde unei direcţii de mutare (regele, dama, calul au 8 direcţii, turnul, nebunul au 4 direcţii etc.). La piesele cu bătaie lungă, incrementul pe o direcţie se repetă pînă la întîlnirea unei piese sau a marginii tablei.

O poziţie pe tablă este, de fapt, un indice în TABLA (120), iar incremenţii se referă la incremenţii de indice.

Cele mai mari dificultăţi în programare le pune pionul, ţinînd seama de neomogenitatea regulilor sale de mutare (transformare, „en—passant", mutare pion 2 paşi etc.).

Ceva dificultăţi creează şi tratarea rocadelor, dar aceste mutări speciale se pot asocia mutărilor regelui. Pentru a rezolva problema detecţiei dacă regele care mută prin rocadă nu trece prin cîmpurile atacate de adversar, unele programe, înainte de a genera lista mutărilor dintr-o poziţie dată, calculează tabele paralele cu tabla de şah pentru a măsura „tăria" cu care cei doi jucători controlează cîmpurile tablei de şah. Aceste informaţii sînt utile pentru acordarea **ponderilor (notelor) statice** ale mutărilor.

Scrieţi subprogramul de generare a listelor de mutări din poziţii date, astfel încît să fie cît mai uşor de modificat, ţinînd cont că acest subprogram va suferi cele mai dese schimbări în procesul evoluţiei programului dv.

Arborele mutărilor.

Pentru a înţelege cum procedează un program de şah cînd calculează **mutarea optimă,** va trebui să facem o descriere a ceea ce înseamnă **arborele mutărilor.**

Într-o poziţie dată pe tabla de şah, ştiind cine este la mutare, se poate genera lista mutărilor posibile la acea mutare. Dacă luăm, pe rînd, fiecare mutare din listă şi o efectuăm, obţinem noi poziţii pe tabla de şah, din care jucătorul advers poate genera cîte o nouă listă de mutări. Repetînd procesul de mai multe ori, alternativ pentru alb şi negru, ne dăm seama că mulţimea de poziţii care apar creşte foarte mult, exponenţial, funcţie de numărul de **semimutări** pentru care se face generarea arborelui. Dacă admitem că într-o poziţie de mijloc există circa 33 mutări, atunci chiar şi pentru 3 mutări (6 semimutări) se generează un arbore cu un miliard de **noduri,** ceea ce este mult chiar şi pentru calculatoarele rapide, căci un nod nu poate fi analizat doar cu o singură instrucţiune a calculatorului.

Să rămînem, deocamdată, în limitele ficţiunii, presupunînd că un calculator poate fi oricît de rapid. Cît de lungi pot fi ramurile în arborele de joc? Se ştie că, prin repetări de mutări, se poate muta la nesfîrşit. Regulamentul jocului de şah asigură însă ca arborele de joc să fie finit, căci sînt **remize** repetările de 3 ori a poziţiei, sau a lipsei unui schimb (sau a mutării unui pion) în mai mult de 50 de mutări.

Evident, **nodurile terminale** în arborele de joc sînt poziţiile de **mat** (regele advers **e în şah**) şi orice ar muta va fi luat la următoarea mutare) sau **pat** (regele advers **nu e în şah,** şi orice ar muta va fi luat la următoarea mutare).

Să remarcăm că, potrivit definiţiei de mai sus, pentru a vedea dacă o mutare este **mat** sau **pat**, mai este necesar să generăm încă două niveluri de semimutări, pînă cînd regele este luat efectiv. Prin urmare, pentru a rezolva un mat în 2 mutări, va trebui să generăm un **arbore (graf) de joc,** de 5 semimutări (3 + 2).

În figura 3 am reprezentat o poziţie în care albul este la mutare şi dă **mat** în 2 mutări.

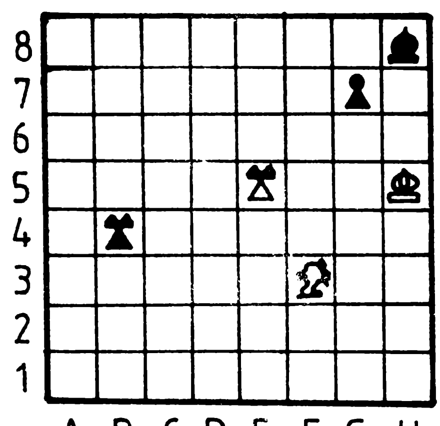


fig. 3 Albul mută. Mat în două mutări.
(alb: rege H5, turn E5, cal F6;
negru: rege H8, turn B4, pion G7

În figura 4 am reprezentat un subarbore al arborelui de joc asociat poziției din figura 3.

Problema detecției faptului că o mutare este **mat** sau **pat**, creează dificultăți în cadrul unui program de șah, dificultăți pe care autorii de programe de șah le rezolvă în fel și chip:

— apelînd subrutine complexe de verificare dacă poziția e mat sau pat;

— prelungind arborele cu încă două semimutări și urmărind dacă regele advers este luat, indiferent ce ar muta;

— tratări mixte, ca de exemplu în programul ATOM—64, în care se fac prelungiri de ramuri în arbore, doar cînd o mutare dă șah.

Cu excepția unor poziții cu puține piese, ori a unor probleme de mat în cîteva mutări, este o utopie să se creadă că un calculator poate analiza toate ramurile pînă la nodurile terminale, asociate unei poziții de pe tabla de șah, chiar dacă s-ar aplica criterii de reducere a variantelor (după cum se va vedea în expunerile viitoare).

Prin urmare, orice program de șah își va face o strategie în privința modului de a-și limita lungimea ramurilor pe care le generează, pentru a se încadra în anumite limite de timp.

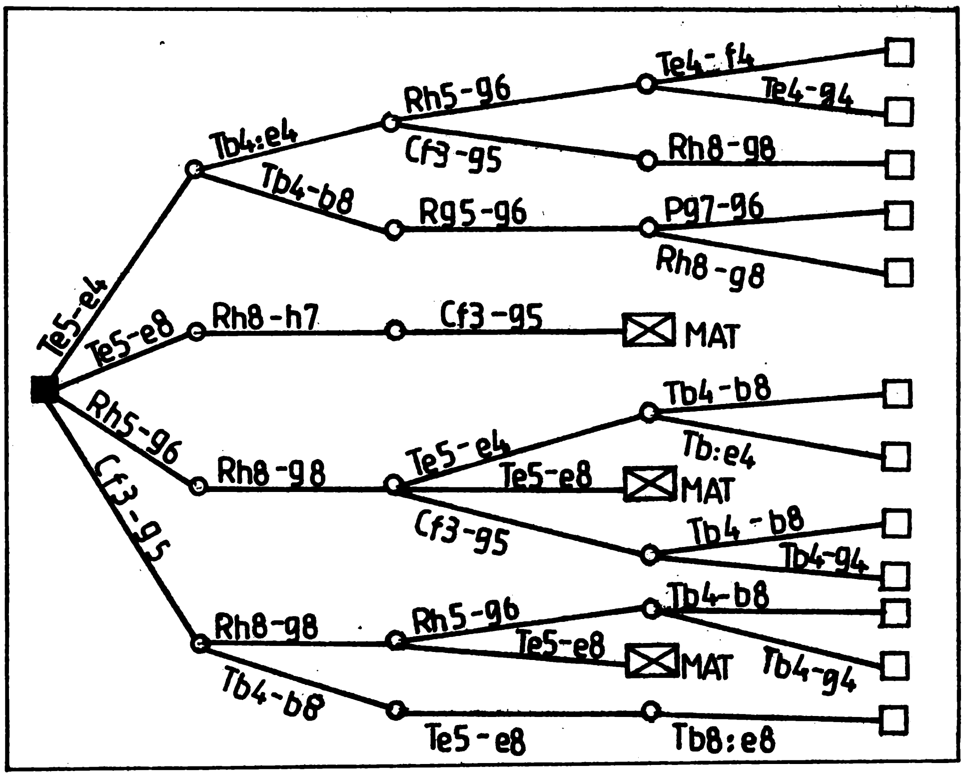


fig.4 O parte din arborele de joc asociat pozitiei din figura 3

Ponderile statice.

Cînd generăm ramurile arborelui de joc, analizăm nodurile terminale, asociindu-le valori, numite **ponderi statice,** adică note pe care le putem calcula fără mutări. Toate remizele sînt la fel de bune, deci ponderile statice asociate lor sînt egale cu zero.

În privința pozițiilor de **mat** însă, orice jucător va muta astfel încît să dea **mat** în cît mai puține mutări și să primească **mat** cît mai tîrziu. Deci, dacă un mat se dă în **x** semimutări, atunci valoarea statică a acelei mutări este A-x (pentru alb) și -A + x (pentru negru). **A** este un număr foarte mare. Cu această precizare ne-am exprimat intenția de a folosi un sistem de **ponderi statice,** în așa fel, încît orice **avantaj** pentru alb să fie adăugat cu semnul „+" la ponderea statică, iar orice **avantaj** pentru negru se ia cu semnul „—". Prin urmare , **penalizările** pentru alb și negru se consideră cu semnul „—", respectiv „+". Deci, albul tinde spre mutări cu ponderi cît mai mari, pozitive, iar negrul tinde spre mutări cu ponderi cît mai negative.

Mutările alternate ale albului și-ale negrului conduc, în cele din urmă, la o luptă de tip **mini-max** , în care învinge cel care reușește să aducă pe adversar în zona convenabilă lui.

Într-un arbore limitat doar de regulamentul jocului de șah, nu am avea decît poziții de **mat** sau **remiză** , deci ponderile lor statice ar fi perfect determinate.

Ce se întîmplă însă într-un arbore cu ramurile trunchiate după un anumit număr de semimutări? Ce pondere statică trebuie acordată acestor poziții?

Răspunsul la această întrebare reprezintă **problema cheie** a teoriei programării jocului de șah. Această problemă nu se poate rezolva îndeajuns de bine decît aplicînd felurite **criterii paleative** care încearcă să **aproximeze** ponderile statice. Pe de altă parte, se constată că, **cu cît ramurile studiate într-un arbore de joc sînt mai lungi, cu atît efectul impreciziei evaluării ponderilor statice se atenuează.** De unde și tendința de a genera arbori cu variante cît mai lungi, fără a le rări însă îmtr-atît încît să se piarda mutarea optimă.

Orice programator de șah se convinge, mai devreme sau mai tîrziu, că nu e bine ca arborele să aibă toate variantele de aceeași lungime. Justificarea acestei afimații constă în aceea că, aprecierea mai bună a **ponderii statice** se poate face în **poziții liniștite,** adică poziții în care nu sînt piese importante atacate (rege în șah, schimb de piese, fuga din șah, amenințări de furculițe, transformări etc.). De asemenea, dacă pe traseul unei variante au existat mutări forțate (schimburi, șah-uri) atunci variantele respective se cer prelungite cu numărul corespunzător de semimutări, căci altfel noile acțiuni gîndite (după respectivul schimb, șah succesiv) nu sînt matur judecate.

Evident, există tentația de ramificare cît mai bogată! Totuși programul trebuie să se încadreze în limita de timp alocată.

Algoritmul mini-max.

Ce este mini—max? Este un procedeu matematic prin care se determină o **variantă optimă** în arborele de joc, ca rezultat al deciziilor de luptă ale ambilor jucători.

Prima mutare dintr-o **variantă optimă** este **mutarea optimă.** În orice nod în care **albul** este la mutare, el va alege mutarea sa corespunzătoare valorii **maxime a minimelor** pe care negrul le va putea alege ca mutare de răspuns. Cînd negrul este la mutare, el alege mutarea sa corespunzătoare valorii **minime a maximelor** pe care albul le va putea alege ca mutare de răspuns. Prin urmare, un proces **mini-max** este un proces recursiv, pe cîteva niveluri. Practic, procesul de evaluare a variantei **mini-max** este un proces care începe dinspre nodurile terminale ale arborelui de joc, spre baza arborelui.

Să considerăm că lungimea variantei celei mai mari în arbore este N și că în fiecare nod al arborelui se ajunge după **i arce,** adică **i** semimutări. Evident, mulțimea nodurilor arborelui se împarte în clase de echivalență, numite **niveluri,** dacă se consideră din clasa **i** nodurile în care se ajunge după **i** semimutări.

Procesul **mini-max** asociază fiecărui nod din arbore o **valoare mini-max** astfel:

— nodurilor terminale li se atribuie ca valori **mini-max** tocmai **ponderile statice;**

— se pleacă de la nivelul cel mai mare; în fiecare nod al unui nivel se consideră valorile mini-max ale mutărilor ce se generează din acel nod și se face cu ele maximum (minimum), dacă în acel nod la mutare este albul (respectiv negrul). Aceste maxime sau minime se asociază nodurilor corespunzătoare și, în momentul în care se îmbunătățește valoarea de maxim sau de minim, se reține și mutarea căreia îi corespunde, mutare ce se **asamblează cu subvarianta de continuare optimă** a acelei mutări, memorată anterior;

— se trece prin nivelul **i-1** și se repetă procesul mini-max.

Avertizăm cititorii că este mai ușor de păstrat **mutarea optimă** decît **varianta de joc optimal** („best line"). Nu este obligatoriu de menținut **varianta de joc optimal,** dar programele mai evoluate o fac și acest lucru le aduce utilități. Problema selecției acestor **variante de joc optimal** o vom mai relua.

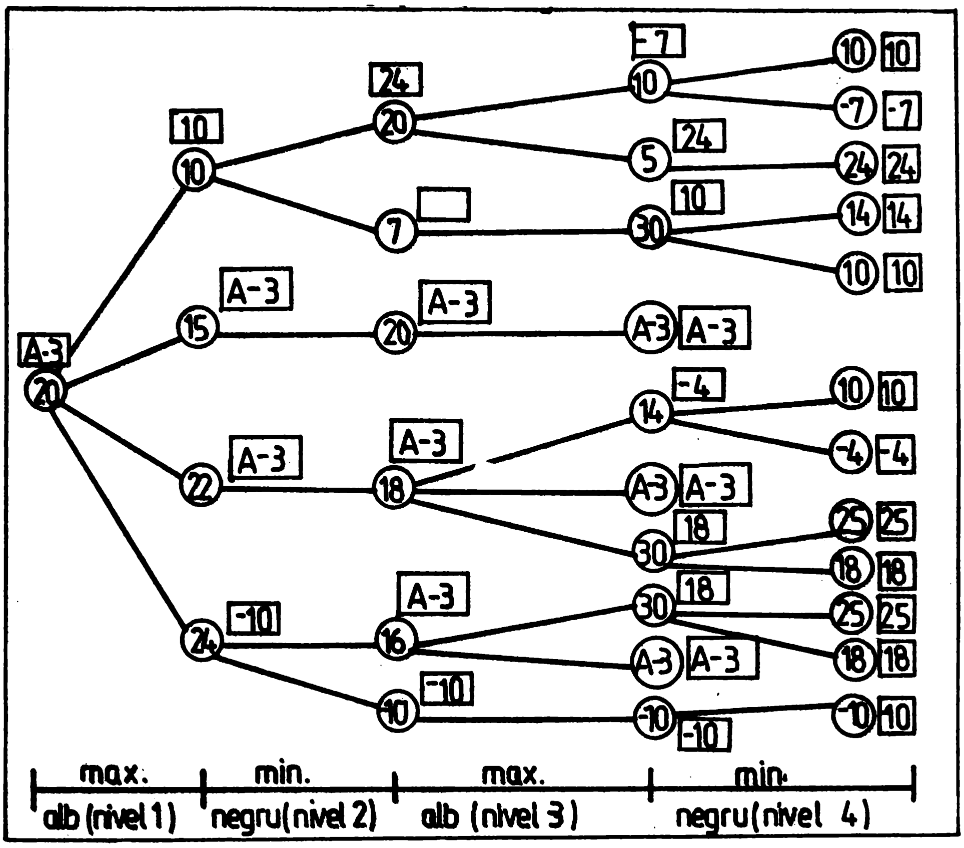


fig. 5 Ponderile statice (în cercuri) şi valorile minimale (în pătrate) atribuite nodurilor arborelui de joc din fig. 4

În figura 5 este prezentat modul în care operează procesul **mini-max**.

Mini-max, pro şi contra.

De ce mini—max? E obligatoriu ca programarea şahului să fie formulată în termeni de variante în arborele de joc şi mini—max?

Am auzit pe unii programatori care nu cunoşteau teoria **mini—max** programînd şah. Evident, ei programau unele judecăţi echivalente cu deciziile de alegere de mutări optime, conform mini-max.

Există obiecţii multe privitoare la modelarea şahului în termeni de arbori de joc şi mini-max. In mod natural, anumite grupuri de piese de pe tabla de şah **interacţionează mai puternic în cadrul grupurilor** şi mai slab între grupuri. Arborele de joc, unic pentru toate piesele, are neajunsul că **proliferează în mod excesiv combinaţii nesemnificative** între piesele diferitelor grupuri. Vă daţi seama de neajunsul acestui fapt cînd lungimile ramurilor sînt limitate la cîteva semimutări.

Această constatare l-a determinat pe M. Botvinik, teoreticianul vestitului program PIONIER, să elaboreze aşa numita teorie a **zonelor de piese** care interacţionează între ele, precum şi un mod de abordare a **combinaţiilor între zone** (deci un sistem pe două niveluri). Din păcate, ideile lui Botbinik, geniale, însă complicate pentru programare, nu s-au impus în lumea şahului programat deoarece sistemul suferea de unele rigidităţi şi imperfecţiuni.

Însăşi ideea de a privi variantele de joc sub forma unui arbore este criticabilă, căci, se ştie, există şi posibilitatea ajungerii în aceleaşi poziţii prin **intervertiri de mutări**. Adică modelul ar trebui să fie un graf orientat şi nu un **arbore**. Totuşi, intervertirile, procentual, nu sînt chiar atît de numeroase pe cît s-ar părea, în plus, gestiunea lor implică un mare consum de spaţiu de memorare şi timp de căutare, încît este mai convenabil să se renunţe la această gestiune.

GENERAREA ARBORELUI DE JOC

Dacă în cele de mai sus am prezentat mai mult teoretic problema arborelui de joc şi a procesului mini—max, acum să descriem cîteva elemente concrete de programare.

Sînt multe modalităţi de abordare a generării de arbori de joc, de alegere a valorilor mini—max şi de aplicare a criteriilor de reducere de variante (alfa—beta, alfa-beta-deep etc.) ce vor fi prezentate în următoarele expuneri. Nu vrem să impunem o conduită de programare (regretabil ar fi să-i îndrumăm pe cititorii noştri pe căi greşite), vrem ca această prezentare să constituie un material de referinţă.

Nu e bine ca un program de şah să genereze întîi de toate variantele de joc (de o anumită lungime) într-o zonă de memorie, ca apoi să înceapă să determine mutarea optimă, conform mini—max. Aceasta deoarece pe de o parte ar fi necesară o zona foarte mare de memorie pentru reţinerea tuturor variantelor, iar pe de altă parte s-ar genera toate variantele, inclusiv cele care n-ar mai fi necesar să fie generate conform unor criterii de reducere de variante.

Cele mai multe programe de şah adoptă o asemenea strategie de generare de variante de joc, încît ramurile să apară în memorie pe rînd, într-o anumită ordine, iar operatorul mini—max să fie aplicat treptat pe fiecare din nivelurile de joc, pe măsură ce anumite ramuri ale arborelui sînt complet analizate. Intr-o structură de liste de mutări (sub formă de stivă) este păstrată în memoria calculatorului **varianta curentă** din arbore ce se generează, precum şi **mutările adiacente** variantei curente.

Mai explicit, pentru fiecare nivel 1,2, 3,... există cîte o listă de mutări L(1), L(2), L(3),... a mutărilor **adiacente variantei curente afcişi o stivă de contexte** asociată variantei curente, în care un element în stivă conţine mai multe informaţii:

— indicator spre începutul listei L(i);

— pointer în interiorul listei L(i), indicînd mutarea din listă pînă la care s-a făcut analiza; (opţional)

— modificări pe tabla de şah pentru a da înapoi mutarea, pentru a ajunge la nivelul **I—I;**

— valoarea mini—max parţială;

— alte informaţii.

In stiva contextelor fiecare pointer spre interiorul a cîte unei liste L(i) indică tocmai mutarea care e făcuta pentru a putea genera lista L(i+1).

Listele L(i), de la un anumit indice **I** încolo, conţin doar mutări ce vor face parte din variantele ce sînt prelungite (schimburi, mutări forţate etc.). Nu are sens să fie trecute, în listele L(i), mutările corespunzătoare nodurilor terminale ale arborelui, **valorile statice** ale acestor mutari terminale se consideră, direct, ca valori **mini—max.**

In momentul în care o **subramură** a grafului de joc este deja analizată, în acel moment se cunoaşte **valoarea mini—max,** precum şi **varianta optimală de joc** corespunzătoare acelei subramuri. Dacă această valoare mini—max îmbunătăţeşte valoarea mini—max parţială pe nivelul imediat inferior, în acest moment se reţine ca variantă optimală de joc varianta obţinută prin concatenarea mutării curente cu varianta optimală de joc a subramurei. Se vede, deci, că pe fiecare nivel este păstrată cîte o variantă optimală (parţială) de joc, care se îmbunătăţeşte pe măsură ce ramurile devin analizate.

De asemenea, momentul îmbunătăţirii valorii unui mini—max pe un nivel, este şi momentul cînd se aplică diferite criterii de reducere a ramurilor în arborele de joc, cînd se observă că ramura analizată conţine o soluţie atît de bună, că nici nu mai are sens să fie studiate ramurile celorlalte mutări din listă, încă neanalizate.

Algoritmul de calcul al mutării optime.

În fine, la sfîrşitul acestei prime părţi a expunerii privind teoria programării şahului, prezentăm algoritmul de generare de variante, de alegere a mutării optime şi a variantei optimale de joc.

Fie M = lungimea propusă a variantelor.

1.Iniţializări: k = 1; a = 0 (a = nr. prelungiri).

2.Ininiţializarea valorii mini—max pe nivelul k.

3. Generarea listei de mutări L(k) în continuarea listei L(k—1).

4. Dacă k este mai mare decît M sau k + a prea mare, sau regele advers e luat, înseamnă că încep să apară no-

duri terminale, deci se testează valoarea mini—max pe nivelul k în raport cu valorile pe nivelele inferioare, pentru a vedea dacă se poate aplica vreun criteriu de abandon al analizei mutărilor din lista L(k).

Dacă criteriul se aplică, se face pasul 6.

Dacă nu, se continuă cu pasul 5.

5. Se alege mutarea cu ponderea statică cea mai bună din lista L(k). Dacă există, mutarea se suprimă din listă, se efectuează pe tabla de şah, ţinînd la zi informaţiile de context pe nivelul k (PLAY).

Salvează şi valoarea a în contextul stivei.

Dacă mutarea e un schimb, un şah, sau răspuns la şah etc., atunci a=a+1. Se face k=k+1 şi salt la pasul 2.

Dacă L(k) devine o listă vidă, se face pasul 6.

6. Valoarea mini—max (complet definită) pe nivelul k se compară cu valoarea mini—max (parţială) pe nivelul k—1. Dacă nu are loc o îmbunătăţire a valorii mini—max, se trece la pasul 9.

7. În cazul îmbunătăţirii valorii mini—max, pe nivelul k—1 se reţine o variantă optimală obţinută prin concatenarea mutării de trecere de la nivelul k—1 la nivelul k, cu varianta optimală de la nivelul k.

8. De asemenea, cînd se îmbunătăţeşte valoarea mini—max pe nivelul k—1 atunci se aplică un criteriu de reducere a variantelor (abandon al analizei mutărilor din lista L(k—1) care au mai rămas) şi se trece la pasul 10. Dacă criteriul nu funcţionează, se continuă cu pasul 9.

9. k=k—1. Dacă k e mai mic decît 0, arborele a fost analizat în întregime, se cunoaşte valoarea mini—max şi varianta optimală de joc (best line).

Dacă k e mai mare ca 0, atunci se revine la contextul valorilor stivei de la nivelul noului k şi se trece la pasul 5 (REPLAY).

10. k=k—1. Se reface contextul stivei la nivelul noului k. Se trece la pasul 9.

COMENTARII

— În limbajele C, PASCAL, care admit apeluri recursive de proceduri, lista L(k), precum şi informaţiile privind un nivel, se pot defini ca variabile locale ale procedurilor, ce se alocă în regim de stivă;

— Testele de la pasul 4 se fac, de fapt, în rutina care genereazǎ lista de mutări L(k), pe măsură ce o mutare este generată;

— Se observă că algoritmul, la pasul 5, nu face întîi o sortare completă a listei L(k), ci doar ia cîte o mutare, cea mai bună din cele rămase în L(k); sînt dese cazuri cînd lista L(k) nu va fi studiată completă (apar reduceri de ramuri), deci o sortare completă a lui L(k) nu este necesară;

— Alegerea mutărilor cu ponderea statică cea mai bună din lista L(k), face ca ramurile arborelui să fie generate în ordinea cît mai plauzibilă de a fi optimale, mutările bune găsite de la început au un pronunţat efect de tăiere a ramurilor adiacente care au mai rămas; astfel, se vede că aprecierea cît mai corectă a ponderilor statice are un rol însemnat atît în acurateţea valorii mini—max stabilite, cît şi a vitezei de analiză a arborelui.

Vom reveni cu alte consideraţii teoretice asupra programării şahului, în special asupra criteriilor de reducere de variante şi de evaluare de ponderi statice.

ANCHETĂ:

În dezbatere: INFORMATICA ÎN ROMÂNIA

Dată fiind importanţa acestui domeniu atît în viaţa socială, cît şi în cea economică, redacţia noastră a iniţiat o anchetă care îşi propune doar să ridice nişte probleme, în concordanţă cu opiniile celor care au avut amabilitatea să ne răspundă, şi nu să dea soluţii definitive. De altfel, ancheta noastră rămîne deschisă, vom mai reveni asupra ei, deoarece subiectul nu poate fi epuizat într-un număr de revistă şi nici nu ne propunem de fapt acest lucru. Scopul nostru final este sensibilizarea unor factori de decizie, deschiderea unor dezbateri constructive, trezirea intereselor — în ultimă instanţă — pentru o disciplină — *informatica* — şi un instrument de lucru — *calculatorul* — fără de care nu poate exista nici o economie modernă. Am mai iniţiat această anchetă şi pentru faptul că ţara noastră a avut matematicieni şi cercetători de frunte care şi-au adus o contribuţie deosebită la fundamentele teoretice ale informaticii şi de ale căror nume se leagă realizări notabile de-a lungul vremii, realizări ce ne-au situat acum vreo trei decenii între ţările cu mari perspective în domeniu. Ceea ce a urmat se cunoaşte şi nu are rost să mai revenim. Important este acum modul în care se abordează acest moment al informaticii româneşti, pentru că această clipă trebuie fructificată la maximum. Şi pe cei reticenţi la astfel de anchete şi, mai ales, la eficacitatea lor îi rugăm să vadă în această anchetă atît sensul ei optimist, de încredere că specialiştii români vor depăşi acest moment dificil şi vor recupera anii ce au trecut, precum şi dorinţa noastră de a sprijini, cu propriile mijloace, dezvoltarea informaticii în România.

Desigur că informatica este în prezent caracterizată printr-o accentuată interdisciplinaritate, deci este puternic condiţionată de aplicaţii. Ca atare, dezvoltarea informaticii în ţara noastră este intim legată şi condiţionată de dezvoltarea economiei în ansamblu. Noi am extras din aceasta *informatica,* pentru a încerca determinarea unor posibile soluţii pe multiple planuri, soluţii menite, în baza aceloraşi legături amintite mai sus, să contribuie la dezvoltarea economiei în ansamblul ei.

Aşadar, am adresat diferitelor personalităţi, cadre didactice universitare, specialişti din institute de cercetări şi din producţia de profil; cadre didactice din învăţămîntul preuniversitar etc., următoarele trei întrebări:

1) În ce direcţii ar trebui dezvoltată producţia românească de calculatoare şi echipamente periferice?

2) Care ar fi direcţiile prioritare de dezvoltare a soft-ului şi dacă acestea ar trebui sau nu să fie în concordanţă cu hard-ul ce se produce în ţară?

3) Care ar fi dotările minimale în şcoli şi de la ce vîrstă pentru cele două direcţii distincte ale învăţămîntului asistat de calculator: informatica în învăţămînt şi informatizarea învăţămîntului?

●

● În ceea ce priveşte răspunsurile la prima întrebare, se impun deja cîteva concluzii, unele sintetizate foarte bine în două dintre răspunsurile primite:

„Vom face adaptări, asimilări ale produselor performante, care sînt cu două generații înainte față de ceea ce se produce la noi, și, ceea ce este poate mai important, să ajungem să posedăm tehnologia la zi pentru aceste produse. Nu cred că trebuie să ne preocupe prea mult producerea calculatoarelor mari — main-frame —, ci producerea de mini și mai ales micro-uri. Pentru utilizări de înaltă profesionalitate trebuie abordat și domeniul micro-urilor dedicate aplicațiilor sub forma stațiilor de lucru. Nu trebuie uitat microcalculatorul ca bun de larg consum, sub nivelul de PC ca preț și posibilități, utilizat în familie pentru divertisment și instruire ca un calculator de casă (home computer) sau ca un joc pe televizor (TV game) care la noi este foarte puțin răspîndit. În privința echipamentelor periferice electronice (terminale, blocuri de măsură electronice etc.), acestea pot fi aduse prin efort intern la performanțe competitive. Pentru perifericele de natură electromecanică (imprimante, plottere, discuri magnetice de diferite tipuri) ar trebui reținute doar acelea la care se poate elabora sau cumpăra tehnologia la zi a părților mecanice".

„Cheia succesului constă în specializarea strictă de nivel cît se poate de înalt. Cred că pentru industria de calculatoare direcțiile principale de dezvoltare trebuie să fie reprezentate de sisteme de calcul medii-mari, de mini și microcalculatoare pe 16 și 32 de biți, cît și de calculatoare personale pe 8 biți. Evident, nu trebuie neglijate nici necesitățile stringente ale informaticii românești la ora actuală privind echipamentele periferice: unități de memorie cu discuri magnetice de tip rigid și flexibil, plottere, imprimante cu laser, stații grafice cu rezoluție ridicată și videoterminale etc."

În general, putem să afirmăm că răspunsurile au fost orientate spre dezvoltarea microcalculatoarelor și a „personalelor" ca bunuri de larg consum; spre programe ușor de utilizat de către neinformaticieni, alinierea la standardele internaționale pentru diferitele familii de calculatoare (IBM, DEC, pentru micro și respectiv minicalculatoare), cooperarea cu firme străine, dezvoltarea rețelelor de calculatoare și a stațiilor de lucru CAD/CAM. În ceea ce privește echipamentele periferice, se desprind cîteva tendințe clare: memorii magnetice (cu capacitate mare — atît pe linia discurilor, cît și a benzilor magnetice streaming), imprimante rapide, display-uri color de înaltă rezoluție.

●

● La cea de-a doua întrebare se desprind de asemenea cîteva concluzii interesante. Axa centrală a majorității răspunsurilor se referă la soft-ul de aplicații care poate să se constituie într-o premisă de bază a unei industrii naționale de soft. Un alt aspect al dezvoltării soft-ului este acela că nu trebuie legat intrinsec de dezvoltarea industriei autohtone, ci poate constitui baza unei activități care să ofere servicii sub forma aplicațiilor diverse. Așadar, „ingineria software poate deveni o componentă extinsă și cu personal foarte numeros în cadrul unei industrii naționale de tehnică de calcul". Aceasta presupune utilizarea programelor standard pentru CAD/CAM etc., care constituie de asemenea un posibil mod de aliniere la cerințele actuale ale pieței mondiale, în diferite domenii. Ar trebui în mod esențial creat cadrul legal care să pună soft-ul în drepturile sale, deoarece el constituie o importantă resursă economică.

O idee interesantă este aceea a „caselor de soft" care se pot dezvolta fie în cadrul institutelor de cercetare și *universităților* (acest lucru ni se pare esențial, și anume implicarea *reală* a studenților și cadrelor didactice în activitatea de cercetare), fie particulare. Ca o posibilă concluzie, citām un fragment dintr-un răspuns:

„Produsele software de aplicație vor constitui un cîmp de activitate cu deschidere largă și care pot forma un segment foarte important al activității pentru o industrie națională de software. Producția de software de anvergură industrială trebuie pornită cît mai repede, deoarece necesită o investiție materială relativ redusă în raport cu valoarea produselor realizate și este tot atît de importantă ca și producția de echipamente".

●

● În legătură cu cea de-a treia întrebare disputele sînt foarte numeroase. Această problematică a mai făcut obiectul a numeroase anchete, deoarece „școala la ora informaticii" se anunța a fi o problemă foarte complexă, cu multiple fațete și cu implicații deosebite; tocmai de aceea, ne-am gîndit ca pentru început să acordăm un spațiu mai mare răspunsului pe care ni l-a dat un cadru didactic din învățămîntul liceal, din care se pot trage foarte ușor concluziile:

„În învățămîntul preuniversitar, informatica trebuie să pătrundă diferențiat:

— în școli profesionale și licee elevii să-și însușească un bagaj minim de cunoștințe care să le permită o adaptare rapidă la tehnica de calcul cu care vor lucra după absolvire;

— să se mențină liceele de informatică, în scopul formării specialiștilor cu studii medii. Aceste licee să funcționeze ca liceu teoretic real cu specialitate informatică.

Necesitatea liceelor de informatică decurge din următoarele motive:

— informatica se învață bine începînd cu vîrsta de 15 ani, cînd puterea de imaginație este maximă;

— schimbările care intervin în domeniul tehnicii de calcul sînt atît de rapide încît nu le poți stăpîni cu adevărat printr-o pregătire particulară de cîteva luni. Însăși programa școlară pentru obiectul informatică trebuie schimbată și actualizată aproape anual;

— specializarea în informatică determină implicit crearea unei culturi generale vaste. Atenția cercetătorilor din domeniul informaticii și profesorilor de informatică, a pedagogilor și psihologilor trebuie concentrată în următoarele direcții: construirea unor «baze de cunoștințe» care să înmagazineze un număr mare și variat de cunoștințe referitoare la diverse domenii, ușor prelucrabile prin dezvoltarea unui software bogat și puternic; utilizarea calculatoarelor în procesul de evaluare a cunoștințelor prin construirea unor teste, pentru diverse discipline de învățămînt".

Deci calculatorul trebuie să pătrundă în școală alături de profesor. Aceasta implică însă mai multe aspecte: o bază materială (deci o industrie care să asigure dotarea *tuturor* școlilor cu calculatoare), adecvată desfășurării în condiții normale a procesului didactic, dotare ce ar trebui să fie diferențiată în funcție de nivelul școlar (microcalculatoare pe 8 biți din gama HC, COBRA etc. în ciclul primar, gimnazii sau unele licee și calculatoare personale pe 16 biți din gama PC XT/AT în licee de profil și facultăți), cadre care să aibă pregătirea necesară predării informaticii, programe care să fie în primul rînd pedagogice și eficiente pentru a întregi actul predării lecțiilor și învățării, disocierea clară a celor două tendințe: informatica în învățămînt — prin menținerea și mărirea numărului de licee de informatică din țară — și informatizarea școlii, folosirea potențialului uman de inteligență al copiilor avînd în vedere rezultatele pe care ei le obțin la competițiile internaționale. Într-o primă etapă, deoarece multe dintre punctele sesizate în răspunsuri presupun rezolvări pe termen destul de lung, cercurile de informatică din școli (la toate nivelurile), patronate de institute de profil și de întreprinderi producătoare, vor pregăti momentul trecerii la informatizarea învățămîntului.

Ancheta noastră nu s-a încheiat. Sintetizînd unele aspecte și neglijînd poate altele, ea nu a dorit decît să deschidă o discuție la care așteptăm în continuare opiniile dumneavoastră.

Anchetă realizată de MIHAELA GORODCOV

Vlad Ţepelea

PROIECTUL ELECT'90

Iată o reuşită, din toate punctele de vedere, a specialiştilor în domeniu din ţara noastră! Menţionăm faptul că materialul ne-a fost remis la redacţie în limba engleză. Noi, din motive multiple, vă vom prezenta doar un rezumat în limba română, iar materialul original a fost trimis prin reţeaua IDG, sub antetul „Contribuţii", la International Data Comunication Group. Pe această cale, invităm pe toţi colaboratorii, cercetătorii şi utilizatorii de sisteme de calcul să ne remită astfel de materiale spre a fi difuzate în reţeaua IDG.

1. Precizări introductive

ELECT'90 este numele proiectului care a avut drept scop procesarea datelor pe calculator referitoare la alegerile din 20 mai 1990 din România. Proiectul a început în martie, după aprobarea legii electorale, şi s-a încheiat pe 25 mai, o dată cu anunţarea oficială a rezultatelor alegerilor.

Derularea proiectului a avut în vedere următoarele: în fiecare capitală de judeţ a fost stabilit cîte un birou local pentru alegeri (District Electoral Office-DEO), în total 41, precum şi un birou central electoral în Bucureşti (Central Electoral Office-CEO). Din partea guvernului a fost însărcinată Comisia Naţională de Statistică - CNS, pentru a duce la bun sfîrşit proiectul. Aceasta, la rîndul ei, a ales ITC-Bucureşti ca proiectant general şi executant pentru partea de software a acestui proiect şi Rom Control Data Ltd. ca producătoare hardware.

Cîteva date statistice: ● 12394 centre de votare în toată ţara; ● în cursa electorală au intrat 71 de partide politice; ● oraşul Bucureşti a fost o excepţie de la Legea electorală cu 1 030 centre de votare, 102 liste de candidaţi pentru funcţiile de deputaţi şi 66 de liste pentru senatori.

2. Hardware

Pentru a duce la bun sfîrşit proiectul ELECT'90 a fost implementată, în timp record, o reţea de nivel naţional. Reţeaua a constat în 41 de noduri în fiecare DEO şi o reţea de tip LAN (Local Area Network) la CEO. Nodurile erau formate din 2-4 PC Compatibile AT (12 MHz, 80286 CPU, 1MB RAM, 3 1/2 floppy disk drive, 40/80 MB hard disk drive şi monitor color EGA), 2 imprimante matriciale, precum şi cel puţin 1 modem inclus.

Reţeaua LAN de la CEO a fost concepută cu două servere de tipul 386 SX/33 MHz cu 330 MB capacitate în hard disc. A fost utilizat pachetul de programe NOVELL 2.15 pentru reţele precum şi adaptoarele LAN ARCNET cu viteză de 2 500 biţi/secundă. Un număr de 7 PC/AT au asigurat comunicaţia la 2 400 bauds, fiecare AT fiind conectat cu 6 judeţe.

Alte două microcalculatoare, de asemenea, compatibile PC/AT, au asigurat prezentarea grafică (parţială şi finală) a rezultatelor,unul dintre acestea fiind în permanenţă conectat la un car TV pentru transmisii. În total, acest proiect a inclus 150 echipamente (calculatoare, imprimante, modemuri etc.). În cadrul CEO s-au utilizat, de exemplu, două imprimante laser de tipul Hewlett Packard, sisteme conectate la server.

3. Software

Proiectul ELECT'90 a avut următoarele scopuri:

— fiabilitate maximă a hard-ului;

— precizie absolută a datelor prelucrate şi a rezultatelor finale sau parţiale;

— protecţia datelor în cazul defectării hard-ului prin salvarea periodică a acestora pe suporturi externe (floppy discuri sau hard discuri);

— interfaţă prietenoasă cu utilizatorul;

— prezentarea grafică a rezultatelor parţiale şi finale pe monitoare TV.

Pentru a mări viteza de introducere a datelor s-au utilizat două terminale compatibile DEC VT 100 cu interfaţă serială.

Funcţiile pachetelor de programare pentru DEO sînt următoarele:

1.— introducerea datelor (cu toate aspectele legate de fiecare birou de votare, număr de electori, voturi acumulate, număr de voturi pentru fiecare listă de candidaţi etc.);

2.— includerea în baza de date a înregistrărilor rapoartelor oficiale;

3.— autentificare;

4.— corecţia bazei de date;

5.— comunicarea datelor-care au presupus transmiterea rezultatelor parţiale şi finale către CEO;

6.— actualizarea conţinutului discurilor între maşinile din cadrului aceluiaşi DEO;

7.— tipărirea datelor şi controlul dispecerului de tipărire;

8.— calculul electoral propriu zis;

9.— verificarea bazei de date.

După colectarea datelor de la fiecare DEO, în cadrul CEO echipamentele specializate în comunicaţii de date, recepţionează rezultatele parţiale şi le transferă la serverul de fişiere prin intermediul LAN, în directorul specific alocat pentru fiecare judeţ, precum şi în directorul special de evidenţă. Periodic, baza de date este împrospătată cu cele mai recente rezultate parţiale din centrele judeţene.

Toate programele de aplicaţie au fost concepute în C, PASCAL şi ASSEMBLER.

Pentru pregătirea personalului implicat în acest proiect s-a organizat un seminar cu durata de 3 zile în Bucureşti.

În ceea ce priveşte atît hard-ul, cît şi pachetele software, pe parcursul derulării proiectului nu au fost înregistrate incidente majore. Echipa formată din tineri specialişti de la ITC, cu o medie de vîrstă de 30 de ani, a demonstrat — pe parcursul a celor 50 de zile, cît a durat proiectul ELECT'90 — că este capabilă de performanţe de nivel european.

În încheiere se aduc mulţumiri tuturor factorilor de decizie şi răspundere, în special Comisiei Naţionale de Statistică, echipei de instalare şi asistenţă tehnică de la Rom Control Data Ltd.; precum şi colegilor de la Centrele Teritoriale de Calcul Electronic şi din ITC pentru efortul depus la ducerea la bun sfîrşit a acestui extrem de interesant proiect, derulat în premieră în ţara noastră şi încheiat cu mult succes în mai 1990.

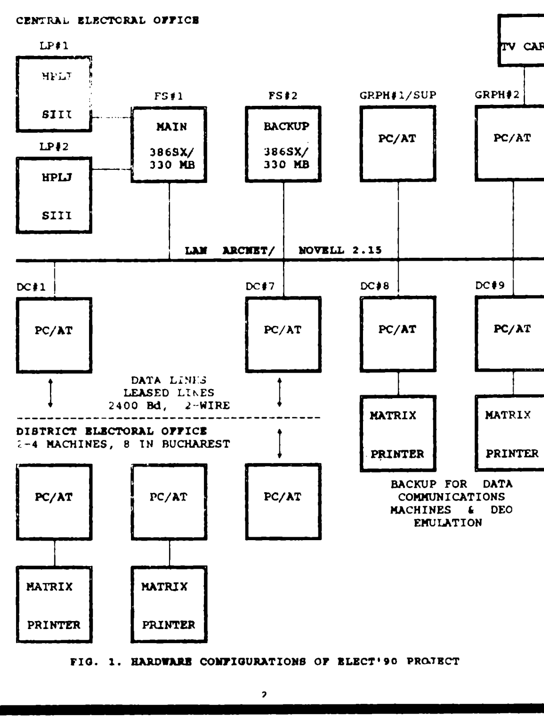


FIG. 1. HARDWARE CONFIGURATIONS OF ELECT'90 PROJECT

Doina ISTRĂTESCU
Eugen GEORGESCU

ELEMENTE PRACTICE DE GRAFICĂ EGA/VGA

Într-un articol din numărul trecut au fost trecute în revistă o serie de principii ce stau la baza cuploarelor grafice utilizate în **calculatoarelor compatibile IBM PC/XT/AT sau PS/2.** Vom încerca acum să aprofundăm aceste noțiuni prin examinarea în detaliu a posibilităților practice de implementare a unor funcții grafice de bază la nivel de coordonate fizice ale cuplorului **Device Coordinates.** Accentul va fi pus pe cuplorul de tip EGA lucrînd în modul 640x350 cu 16 culori simultane. Extinderea la VGA în modul 640x480 cu 16 culori simultane este banală, iar în modul 320x200 cu 256 de culori simultane tratarea problemelor este, în mod surprinzător, mult mai simplă.

Scopul final este ca dumneavoastră să puteți implementa o bibliotecă de funcții grafice. Pentru ușurința lucrului am ales un limbaj de nivel înalt de mare popularitate și anume **C,** astfel încît efortul de programare să fie minim, iar întreaga dumneavoastră atenție să poată fi focalizată asupra problemelor de grafică. Odată depășită etapa de implementare se poate trece la optimizare prin rescrierea unor proceduri în limbaj de asamblare, obținîndu-se astfel sporuri de viteză între 5 și 50%.

Probabil că prima întrebare pe care o puneți este: de ce să construim o nouă bibliotecă în condițiile în care compilatoarele actuale posedă deja una? Motivele sînt două:

— ele nu răspund decît în mică măsură necesităților de realizare a unor efecte grafice nestandard;

— nu asigură instrumente pentru realizarea interactivității cu utilizatorul.

Deoarece cuplorul EGA standard are porturi WRITE—ONLY starea lui este memorată într-o copie RAM a bibliotecii, copie a cărei poziție și structură pot fi cu greu găsite. Orice artificiu realizat direct asupra cuplorului va duce la o incoerență între starea reală și copia RAM, consecințele putînd fi imprevizibile. Deci pentru orice aplicație ce necesită facilități inexistente în biliotecile compilatoarelor este necesar să ne bazăm pe un produs aflat sub controlul nostru total.

În cazul limbajului C, din considerente de dezvoltare și exploatare vom utiliza un grup de mai multe fișiere cu următoarele semnificații: LIB.H — header cu definiri de constante și tipuri; LIB-BAS.C — funcții de control al regimului de lucru; LIB-DC.C — primitive grafice la nivel Device Coordinates; LIB-BLK.C — funcții de lucru cu blocuri de imagine; APL.C — fișier „aplicație" în care vom exersa utilizarea funcțiilor. Acest fișier nu va fi introdus în cadrul bibliotecii (GRF.LIB).

„Aplicația" pe care noi am construit-o pentru dezvoltarea și verificarea funcțiilor grafice a fost un editor grafic de mică complexitate. Pe acesta nu-l vom prezenta pentru a nu abate atenția de la problemele de implementare a funcțiilor grafice pentru cuplorul EGA/VGA. El nu face altceva decît să apeleze funcțiile grafice la comenzi ale unor taste (în regim „hotkeys") cu parametri luați din context sau de la claviatura numerică. Acest lucru îl puteți realiza și dumneavoastră foarte simplu și într-o infinitate de variante.

Pentru a completa contextul de lucru, în cazul în care utilizați compilatorul **Turbo C,** mai este necesar să realizați și un „project file", denumit GRF.PRJ cu următoarul conținut:

```
LIB-BAS.C   (LIB.H)
LIB-DC.C    (LIB.H)
LIB-BLK.C   (LIB.H)
APL.C       (LIB.H)
```

Introduceți numele acestui fișier la opțiunea PROJECT și salvați configurația curentă a compilatorului.

Aceste detalii preliminare fiind lămurite, să trecem efectiv la problemele de grafică.

Primele informații pe care trebuie să le introducem în header sînt cele referitoare la adresa de început a memoriei ecran și adresele porturilor EGA.

LIB.H:

```
#define  EGA_MEM   0xa0000000
#define  P_seq_s   0x3c4      /* Sequencer select */
#define  P_seq_d   0x3c5      /* Sequencer data   */
#define  P_ctr_r   0x3cf      /* Controller register */
#define  P_ctr_a   0x3ce      /* Controller address */
#define  P_atpar   0x3da      /* Attribute/Palette reset */
#define  P_at_pa   0x3c0      /* Attribute/Palette */
#define  P_crt_a   0x3d4      /* CRT Controller Address */
#define  P_crt_r   0x3d5      /* CRT Controller register */
#define  RES_X     80         /* Nr. bytes/line */
```

Asupra ultimei constante dorim să atragem atenția că oricare ar fi modul de lucru (alfanumeric sau grafic) totdeauna pe o linie sînt baleiați 80 de octeți (excepție: modul 13H la VGA).

Prima funcție pe care trebuie să o implementăm este aceea de trecere în mod grafic. Practic, înseamnă a programa porturile cuplorului cu valorile cuprinse în PROM-ul adițional. Cea mai simplă metodă este să utilizăm direct o funcție BIOS. În același timp e bine să schimbăm și setul de caractere la versiunea 8 x 8 (43 de linii). De asemenea, tot în cadrul acestei funcții vom defini și inițializa un pointer deosebit de important a cărei utilitate se va vedea ulterior.

ACTUALITATEA PC



```
LIB_BAS.C:
#include   <dos.h>
#include   "lib.h"
union      REGS  *preg,reg;
char far   *p_ega;

gopen()

{
preg = &reg;
preg->x.ax = 0x0010;
int86(0x10,preg,preg);    /* 640*350 */
preg->x.ax = 0x1123;
preg->h.bl = 0x03;
preg->h.dl = 43;
int86(0x10,preg,preg);    /* 43 linii 8*8 */
p_ega = MK_FP(0xa000,0);
}
```

Tot în acest fișier se găsește și funcția complementară care asigura revenirea în mod alfanumeric:

```
LIB_BAS.C:
gclose()
{
preg->x.ax = 0x0002;
int86(0x10,preg,preg);
}
```

O problemă ce a fost omisă în funcția gopen( ) este aceea a depistării cuplorului EGA/VGA. Soluțiil sînt multiple, dar o versiune simplă este să verificăm dacă la adresa 0000:0487 avem marca 0x60.

CULORI

Odată intrați în mod grafic ne vom gîndi imediat la implementarea unor primitive grafice: line, polyline, circle, rectangle etc. Înainte de aceasta trebuie să ne aducem aminte că ele sînt caracterizate printr-o serie de atribute, dintre care cel mai important este culoarea. Dacă vom reciti articolul din numărul trecut vom vedea că stabilirea unei culori de trasare se poate face în două moduri:

— stabilirea unui index în LUT (Look-Up-Table);

— stabilirea proporțiilor RGB pentru indexul curent în LUT.

Deoarece primitivele sînt diferite, iar unele au mai mult de un atribut de culoare, așa cum vom vedea mai tîrziu, nu putem să realizăm atacul direct al porturilor cuplorului. Vom atașa fiecărei primitive grafice o imagine memorie a atributelor de culoare pe care o vom folosi pentru programarea cuplorului numai în momentul trasarii. Deci setarea unui atribut de culoare se reduce la înscrierea unei variabile interne a bibliotecii, transparenta pentru utilizator.

Al doilea mod de stabilire a culorii de trasare atacă direct cuplorul, modificînd procentele RGB la indexul de intrare în LUT preluat din variabilele interne. Astfel pixelii de pe ecran trasați cu un anumit index își vor modifica culoarea fizică (vezi în numărul trecut legătura între vuloarea logică - index - și fizică - RGB - stabilită prin intermediul LUT). Pentru a putea păstra o compatiblitate între EGA și VGA valorile RGB nu vor fi date în binar ci vor fi normalizate 00...10. Funcțiile pentru VGA pot fi ușor adaptate la o paletă de 256 k culori.

Se observă că din același motiv al imposibilității citirii porturilor EGA (modele mai vechi) am introdus și o copie în memorie a LUT (Look-Up-Table).

Pentru cei ce doresc în mod efectiv să implementeze pas cu pas rutinele din acest articol atragem atenția că **declarațiile variabilelor globale trebuie plasate înaintea declarațiilor de funcții și declarate externe în celelalte module.**

Următoarele funcții realizează efectiv comanda porturilor EGA/VGA și dacă este să ne păstrăm în standardul CGI (Computer Graphics Interface) ele trebuie să fie interne bibliotecii (transparente utilizatorului).

```
LIB_BAS.C:
unsigned char color,bk_color,per_color;
unsigned  char LUT[16];
background_color_index(index)
     unsigned  char index;
     {
     bk_color = index;
     }

background_color(red,green,blue)
     float      red,green,blue;
     {
     set_LUT_color(bk_color,RGB_to_mask(red,green,blue));
     LUT[bk_color] = RGB_to_mask(red,green,blue);
     }

line_color_index(index)
     unsigned  char index;
     {
     color = index;
     }
```

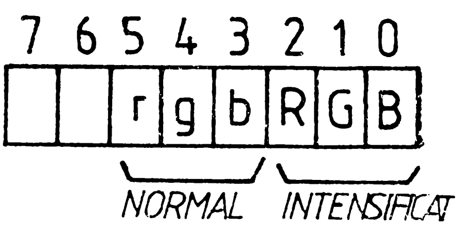
```
line_color(red,green,blue)
     float      red,green,blue;
     {
     set_LUT_color(color,RGB_to_mask(color,red,green,blue));
     LUT[color] = RGB_to_mask(color,red,green,blue);
     }
```

Acesta este punctul în care VGA poate oferi incomparabil mai multe posibilități, și anume o paletă de 256 k culori! Aceasta se realizează în modul 12H, mod în care următoarele rutine devin specifice VGA și mult mai complicate. Din aceste considerente de parc de calculatoare existent în România, ne vom rezuma totuși la un set de rutine compatibile mod 10H EGA (640x350) cît și mod 12H VGA (640x480) în care vom afișa 16 culori simultan dintr-o paletă de 64.

```
LIB_BAS.C:
set_LUT_entry(index)
 unsigned char index;
 {
 outp(P_ctr_a,0x01);
 outp(P_ctr_r,0xff);
 outp(P_ctr_a,0);
 outp(P_ctr_r,index);
 outp(P_ctr_a,0x01);
 outp(P_ctr_r,0x00);
 outp(P_ctr_a,0x00);
 outp(P_ctr_r,0x00);
 }
 set_LUT_color

 (index,mask)
 unsigned  char
 index,mask;
 {
 inp(P_atpar);
 outp(P_at_pa,
 index & 0x0f);
 outp(P_at_pa,mask);
 inp(P_atpar);
 outp(P_at_pa,0x20);
 }
```

Următoarea rutină de conversie de la valori normalizate RGB la mască de biți se bazează pe următoarea structură a unei intrări în LUT:




```
LIB_BAS.C:
RGB_to_mask(red,green,blue)
     float      red,green,blue;
     { unsigned      char mask;
     mask = 0x00;
     if(red <= 0.25)
          ;
     else if(red <= 0.5)
          mask |= 0x20;
     else if(red <= 0.75)
          mask |= 0x04;
     else
          mask |= 0x24;
     if(green <= 0.25)
          ;
     else if(green <= 0.5)
          mask |= 0x10;
     else if(green <= 0.75)
          mask |= 0x02;
     else
          mask |= 0x12;
     if(blue <= 0.25)
          ;
     else if(blue <= 0.5)
          mask |= 0x08;
     else if(blue <= 0.75)
          mask |= 0x01;
     else
          mask |= 0x09;
     return mask;
     }
```

Culoarea nu este însă un atribut doar al background-ului și foreground-ului. Pentru cuploarele EGA/VGA în afara ariei de pixeli există un chenar a cărui culoare nu este legată de nici o intrare în LUT și poate fi specificată direct. Privind în corespondență cu semnalul TV ea corespunde zonei de blanking și în cazul nostru poartă denumirea de overscan.

```
LIB_BAS.C:
overscan_color(red,green,blue)
float      red,green,blue;
{
inp(P_atpar);
outp(P_at_pa,0x11);
outp(P_at_pa,RGB_to_mask(red,green,blue));
inp(P_atpar);
outp(P_at_pa,0x20);
}
```

LINIA

Vom trece acum la delicata problemă a trasării unei primitive grafice. Pentru a înțelege mai întîi mecanismele de utilizare a registrelor EGA/VGA vom studia un caz unde nu intervine nici un algoritm de trasare și anume „aprinderea" unui pixel pe ecran. Aceasta implică două probleme:

— setarea culorii;

— stabilirea corespondenței între poziția geometrică exprimată prin coordonatele (x, y) și adresa fizică a memoriei de afișare.

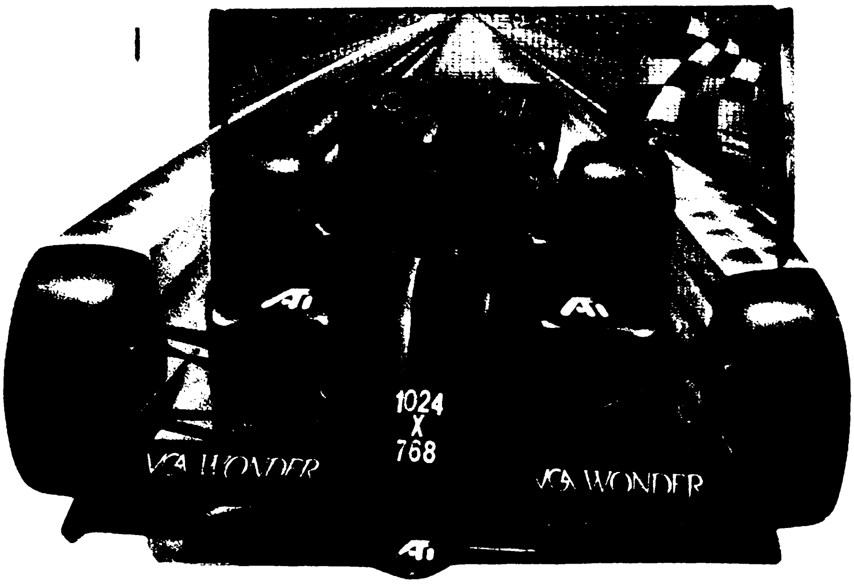
Așa cum am arătat în numărul trecut, memoria video are o structură spațială. Procesorul calculatorului nu are acces decît pe suprafața x, y și chiar și acolo numai trecînd prin ALU și o mască. Pentru lucrul în adîncime, deci cu indexul de culoare, nu se lucrează decît prin intermediul unei perechi de registre care se găsesc la index 0 și 1 în Graphics Controller. În registrul 0 se scrie practic culoarea logică de trasare, adică codul uneia din cele 16 intrări în LUT. În momentul activării unui pixel, în fiecare plan de culoare se înscrie bitul corespunzător din acest registru. În realitate situația este puțin mai complicată fiindcă apare o validare suplimentară prin registrul 1. Astfel, un 1 în registrul de validare va permite înscrierea planului cu valoarea corespunzătoare, iar un 0 în registrul de validare va lăsa conținutul neschimbat din punctul de vedere al culorii.

Nici cea de-a doua problemă nu este simplă. Din punct de vedere geometric un pixel se găsește într-un spațiu cu două dimensiuni. În marea majoritate a modurilor grafice ale EGA/VGA unui pixel îi corespunde în fiecare plan de culoare un bit. Din nefericire bitul se găsește într-un spațiu cu alte proprietăți organizat unidimensional (linear de la adresa a000:0000) și discretizat în cuante numite octeți (bytes).

Deci înainte de orice operație de trasare trebuie să efectuăm o transformare din spațiul geometric de coordonate (x, y) în spațiul de memorie byte (relativ la începutul memoriei ecran) și poziție în interiorul byte-ului și să lucrăm folosind acest nou sistem de coordonate. Din păcate nu este suficient fiindcă în Graphics Controller se mai găsește un registru cu numărul 8. În el se înscrie o mască ce indică biții care pot fi modificați în interiorul unui byte. Rolul său este foarte important și anume acela de a păstra nealterate elementele grafice anterior trasate; cu alte cuvinte, a doua coordonată denumită „poziția în interiorul byte-ului" trebuie transformată în mască. Din punct de vedere software soluția cea mai simplă este utilizarea unui tabel de măști.

Înainte de a exemplifica programul mai avem de făcut o observație. Așa cum s-a văzut în schema funcțională a cuplorului prezentată în numărul trecut, scrierea datelor de la procesor în memoria video se face printr-un ALU care trebuie alimentat cu vechea valoare printr-o operație de citire.

În limbajul C, cele două operații de citire și scriere pot fi condensate printr-un artificiu folosind operatorul „|=" care nu perturbă funcția selectată în ALU.




```
LIB_DC.C
unsigned    char pxmask[8] =
     {0x80, 0x40, 0x20, 0x10, 0x08, 0x04,
                              0x02, 0x01};
put_pixel(x,y,pixel_color)
     int  x,y;
     char pixel_color;
     {int byte;
      outp(P_ctr_a,0x01);
      outp(P_ctr_r,0xff);
      outp(P_ctr_a,0x00);
      outp(P_ctr_r,pixel_color);
      byte = (x >> 3) + RES_X*y;
      outp(P_ctr_a,8);
      outp(P_ctr_r,px_mask[x & 7]);
      * (p_ega+byte) |= 0xff;
      outp(P_ctr_a,8);
      outp(P_ctr_r,0xff);
      outp(P_ctr_a,0x01);
      outp(P_ctr_r,0x00);
      outp(P_ctr_a,0x00);
      outp(P_ctr_r,0x00);
     }
```

Transformarea din sistemul de coordonate (x, y) în sistemul (byte, poziție în byte) s-a făcut prin relațiile:

```
byte    int(x/8) + yx(nr_bytes/linie)
pozit   = rest (x/8)      sau    x modulo 8
```

Deoarece 8 este egal cu 2 la puterea a treia, pentru creșterea substanțială a vitezei, împărțirea întreagă cu 8 s-a făcut prin „shiftarea" dreapta cu 3 poziții, iar restul aceleiași împărțiri nu este altceva decît numărul cuprins în cei trei biți mai puțin semnificativi care se izolează prin mascarea cu 7.

Se mai poate observa că la terminarea operațiunii de trasare a pixelului, registrele cuplorului se readuc

într-o stare cunoscută care s-a ales ca fiind cea standard din inițializare.

Am insistat în mod deosebit asupra „problemei pixelului" fiindcă prezintă dificultăți pentru cei ce au la dispoziție doar documentația porturilor EGA/VGA și în plus este esențială pentru înțelegerea următoarelor primitive grafice. În cazul în care se dorește utilizarea VGA în modul 13H (320x200 cu 256 culori simultane) problema se banalizează fiindcă fiecărui pixel îi corespunde un byte, iar culoarea logică este dată de valoarea conținută în acesta.

Problema imediat următoare trasării unui pixel este aceea a trasării unei linii. Această problemă are două aspecte:

— clasele de funcții pentru trasarea liniei;

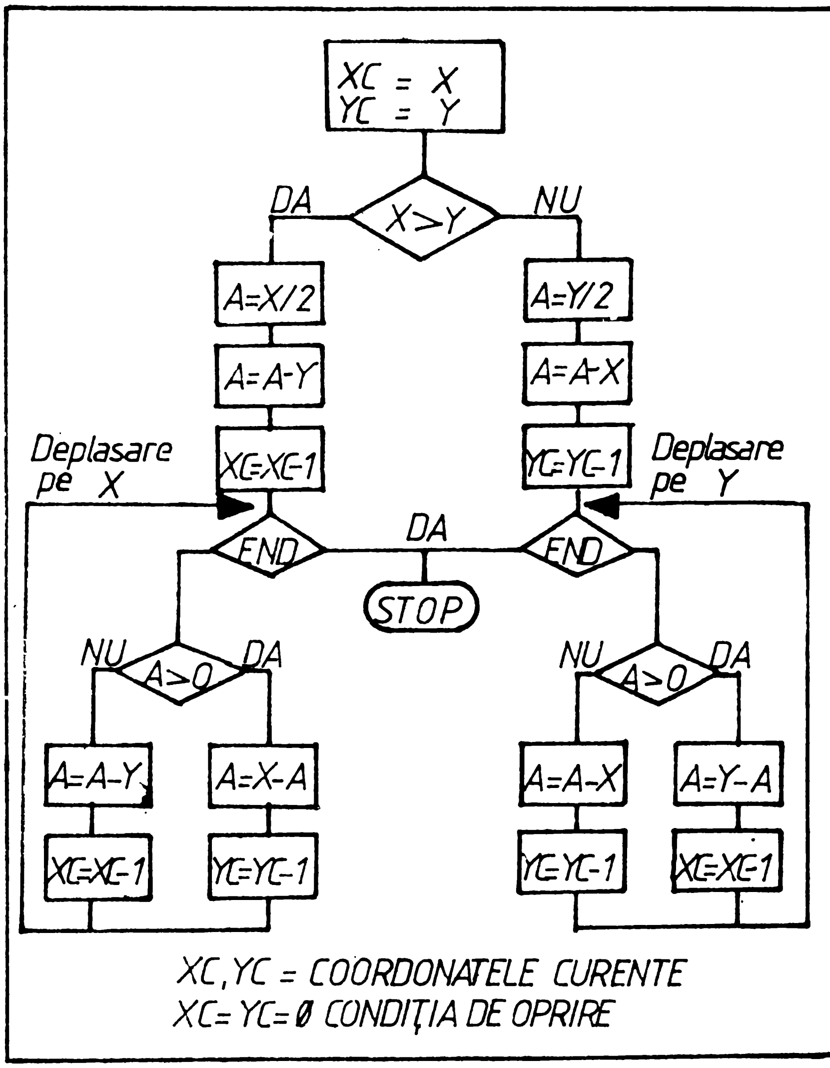
— algoritmul de trasarea liniei.

Există două clase mari de linii ce pot fi trasate:

— linii între coordonate absolute, la acestea precizîndu-se perechile de coordonate (x, y) pentru început și sfîrșit. Funcția din această clasă va purta denumirea de **line ( )**.

— linii care se trasează între ultima coordonată a unei trasări anterioare și o pereche de coordonate (x, y) specificată. Aceasta induce necesitatea memorării coordonatelor finale ale trasării anterioare într-o pereche de variabile interne ale bibliotecii. Funcția din această clasă poartă denumirea de **draw( )**. Aceasta atrage după sine funcția **move( )** care are rolul de a muta originea trasării relative.

Algoritmii de trasare a unei linii pentru display-urile de tip raster nu sînt prea numeroase și vom exemplifica algoritmul Bresenham care este extrem de popular prin raportul performanțe/complexitate. Avantajul său major este că lucrează direct cu ecuația dreptei, ceea ce ar necesita operații lente de înmulțire/împărțire. El transformă totul într-un proces iterativ, bazat numai pe operații de adunare/scădere. Se avansează din pixel în pixel pe axa x sau y după un test asupra pantei și diferențialelor componentelor sale verticale și orizontale. Deoarece sursa reprezintă o extindere pentru patru cadrane, vom prezenta pentru o mai ușoară înțelegere și organigrama pentru o trasare între punctele (0,0) și (x,y).



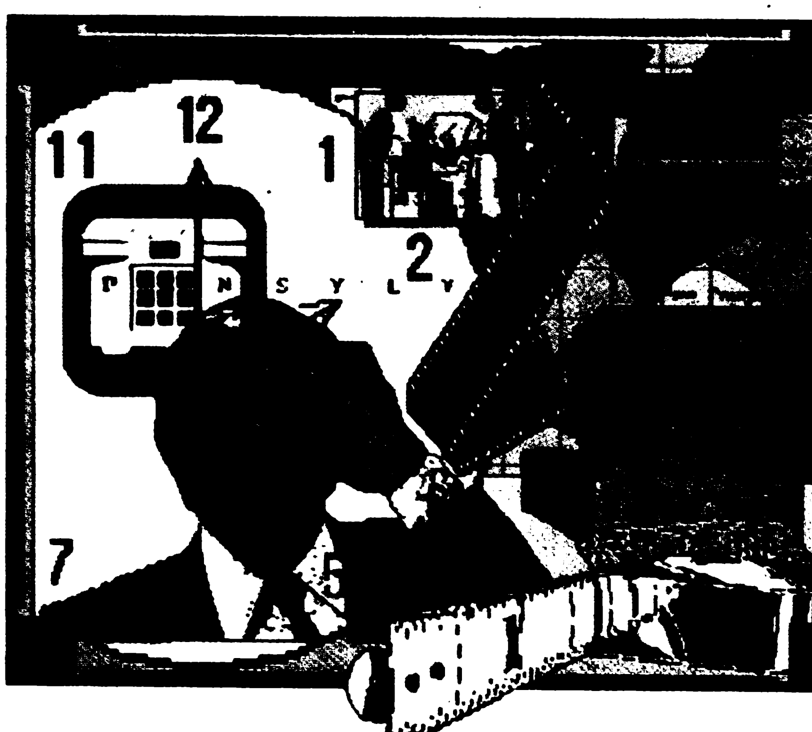
În clipa în care vom implementa acest algoritm vom observa că linia pe care o dorim noi va fi fragmentată într-o serie de segmente pe verticală sau pe orizontală, funcție de pantă. În literatura de specialitate acest fenomen poartă denumirea de **aliasing**. Există algoritmi perfecționați anti-aliasing, dar aceștia dau două tipuri de efecte secundare:

—o grosime minimă a liniei mai mare de un pixel (ex. IBM 5080);

— deși rezoluția este mare, se lucrează cu un dispozitiv virtual cu rezoluție cel puțin înjumătățită (ex. HP 9000/seria 300).

În concluzie acest efect neplăcut nu poate fi compensat decît prin rezoluție mare. La limită putem apela la 800x600, mod nestandard la VGA, dar monitorul trebuie să aibă frecvența de baleiaj pe orizontală de 40 kHz, ceea ce ne va costa minim încă 2—3oo dolari.

Să analizăm acum în limbajul C:



```
unsigned int last_x,last_y;

line(x1,y1,x2,y2)
unsigned int x1,x2,y1,y2;
{outp(P_ctr_a,0x01);
 outp(P_ctr_r,0xff);
 outp(P_ctr_a,0x00);
 outp(P_ctr_r,color);
 b_line(x1,y1,x2,y2);
 last_x = x2;
 last_y = y2;
 outp(P_ctr_a,1);
 outp(P_ctr_r,0);
 outp(P_ctr_a,0);
 outp(P_ctr_r,0);
}

draw(x,y)
 unsigned int x,y;
 { outp(P_ctr_a,1);
   outp(P_ctr_r,0xff);
   outp(P_ctr_a,0);
   outp(P_ctr_r,color);
   b_line(last_x,last_y,x,y);
   last_x = x;
   last_y = y;
   outp(P_ctr_a,1);
   outp(P_ctr_r,0);
   outp(P_ctr_a,0);
   outp(P_ctr_r,0);
 }

move(x,y)
 unsigned int x,y;
 { last_x = x;
   last_y = y;
 }
```

```
b_line(xp,yp,xf,yf)
unsigned xp,yp,xf,yf;
{register int iv,px,py;
 int byte,nx,ny,nt,nd,ns;
 unsigned int anx,any;
 nx = xf-xp;
 ny = yf-yp;
 px = (nx > 0) ? 1: -1;
 py = (ny > 0) ? 1: -1;
 anx = abs (nx);
 any = abs (ny);
 if(anx > any)
     {nt = anx;
      nd =any;}
 else
      nt = any;
      nd = anx;}
ns = nt - nd;
iv = (nt >> 1) - ns;
while(nt)
{byte = (xp>>3) + RES_X*yp;
 outp = (P_ctr_a,8);
 outp = (P_ctr_r,pxmask[xp & 7]);
 *(p_ega+byte) |= 0xff;
 nt--;
 if(iv >= 0)
     {iv -= ns;
      xp += px;
      yp += py;}
 else
     {iv += nd;
      if(anx > any)
          xp += px;
      else
     }    yp += py;
     }
outp = (P_ctr_a,8);
outp = (P_ctr_r,0xff]);
}
```

Codul nu este scris deosebit de elegant fiindcă s-au făcut toate artificiile necesare pentru creșterea vitezei. Cei cîțiva octeți pe care veți remarca că îi vom „strica" la diverse funcții pentru măști precalculate sau scoaterea unor apeluri de funcții în afara buclelor, vor aduce un „profit" deloc neglijabil sub aspectul vitezei.

Deși există riscul să vă pierdeți răbdarea, nu putem încă părăsi domeniul banalei linii deoarece ea mai are încă atribute specifice.

Unul din ele este pattern-ul cu care se trasează linia, deci o mască care să acopere ciclic pixelii. Soluția este banală și constă într-un AND al măștii registrului 8 al P-ctr cu biții succesivi ai pattern-ului. Deoarece în limbajul C nu avem operații de rotație (ci numai de shiftare) vom apela la un artificiu și anume refolosirea setului de măști prestabilite, **pxmask**, cu indexul preluat după un contor. Nu este necesar să impunem un nou contor, el există, și anume variabila nt.

Atenție! Înainte de a modifica funcția b-line ( ) (Bresenham line) faceți cu editorul o copie numită e-line ( ) (echo line). Această nouă funcție ne va fi necesară la trasarea unor cursoare grafice, care trebuie să acționeze în timp real, nu au atribut de pattern și trebuie să poată fi trasate cu maxim de viteză.

Atributul de pattern de linie, va fi o variabilă internă bibliotecii și va fi introdus cu o funcție specifică.

(continuarie în pag.27)

```
LIB_BAS.C
unsigned        char    style = 0xff;

line_style(pattern)
          unsigned        char    pattern;
          {style = pattern;
          }
```

D.P. Tiberiu

HERCULES II

Impactul microcalculatoarelor personale compatibile IBM a fost foarte mare şi datorită facilităţilor grafice pe care le oferă acestea în diferite opţiuni.

Iată cîteva dintre ele:

- **CGA — Color Graphic Adapter (definiţie 640x240 pixeli);**
- **PGC — Professional Graphic Controller;**
- **MDA — Monochrome Display/Printer Adapter (80 semne, 25 rînduri, matrice 8x14 pixeli);**
- **MGC — Monochrome Graphic/Printer Card**
- **HGC — Hercules Graphic Card; producător Hercules Computer Technology.**

Hercules poate lucra în două moduri: • mod grafic: definiţie 720x348 pixeli • mod text: 80x25, matrice 9x14 pixeli

— frecvenţă de baleiere: • orizontală 18,52 kHz • verticală 50 Hz

Denumirea de Hercules este marcă înregistrată a firmei Hercules Computer Technology. De multe ori versiuni echivalente pot avea alte denumiri cum ar fi, de pildă, MGA (Monochrome Graphics Adapter) şi altele, deoarece denumirea de „Hercules" este protejată.

Placa HGC — în varianta a II-a este executată cu circuite integrate de tip MSI (medium scale of integration), iar următoarele versiuni, mai ales HGC IV au cîştigat mult în densitate prin folosirea circuitelor specializate de tip LSI. Descrierea ultimei versiuni (cea de-a patra) ar fi foarte laborioasă şi dificilă, motiv pentru care, în cele ce urmează, vom face o prezentare mai amplă a versiunii a II-a.

Modul text. Imaginea este plasată în memorie în zona 0B0000H-0B0FFFH (4 KB). Fiecare semn este însoţit de 2 bytes, reprezentînd codul ASCII şi respectiv atributele. Codurile de definire a semnelor sînt memorate secvenţional cu 160 bytes pe rînd.

Adresa unui semn plasat în coloana X (în intervalul 1—80), şi rîndul Y (în intervalul 1—25), faţă de adresa de bază se poate calcula astfel:

160x (Y−1) + 2x(X−1)

Generatorul de semne permite obţinerea de 256 simboluri (specificate de codurile Ascii) diferite. A doua diferenţiere — atributele care sînt următoarele: underline, blank, bold face (deschis la culoare), blink sau reverse video.

Modul grafic se caracterizează prin faptul că fiecare bit reprezintă un punct de pe ecran. Schimbarea conţinutului punctului (X, Y) cu $X \in 0{,}719$ şi $Y \in 0{,}347$, modificat faţă de adresa de bază a paginii 0—0B0000H sau 0B8000H pentru pagina 1 se calculează astfel:

2000 Hx (Y mod. 4) + 90 x INT (Y/4) + INT (X/8)

Locul bitului în byte-ul care conţine punctul (X,Y) este: 7−(X mod 8).

Cea mai comodă este folosirea procedurilor MS-DOS, mai ales în modul text. În modul grafic, HERCULES a lansat o dată cu opţiunea HGC şi programele HBASIC şi Graph X. În acest ultim pachet se regăsesc 15 proceduri grafice elementare (primitives) operate în mod maşină sau într-un limbaj de nivel superior. Programele LOTUS 1-2-3, Turbo Pascal, Auto CAD sînt dotate cu proceduri proprii de comandă a imaginii grafice (drivers) şi cu algoritmi de desenare rapidă.

Modul de codificare a atributelor. Poziţia bitului:

7	6	5	4	3	2	1	0	Atribut
B	0	0	0	I	0	0	0	(blank)
B	0	0	0	I	0	0	0	(underline)
B	0	0	0	I	1	1	1	(normal video)
B	1	1	1	I	0	0	0	(reverse video)

I = 1 deschis la culoare

B = 1 dacă $b_5 = 0$ fond deschis şi $b_5 = 1$ pîlpîire

Comanda display-ului. Comanda monitorului monocromatic se face cu ajutorul a 5 registre (la adresele indicate în paranteze):

- Index Register (03B4H);
- Data Register (03B5H);
- Display Mode Control Port (03B8H);
- Display Status Port (03BAH);
- Configuration Switch (03BFH);

Spot

HERCULES II

1 BLOC MEMORIE IMAGINE

2 BLOC FORMARE IMAGINE

3 BLOC PARAMETRI IMAGINE

4 MEMORIE ATRIBUTE IMAGINE

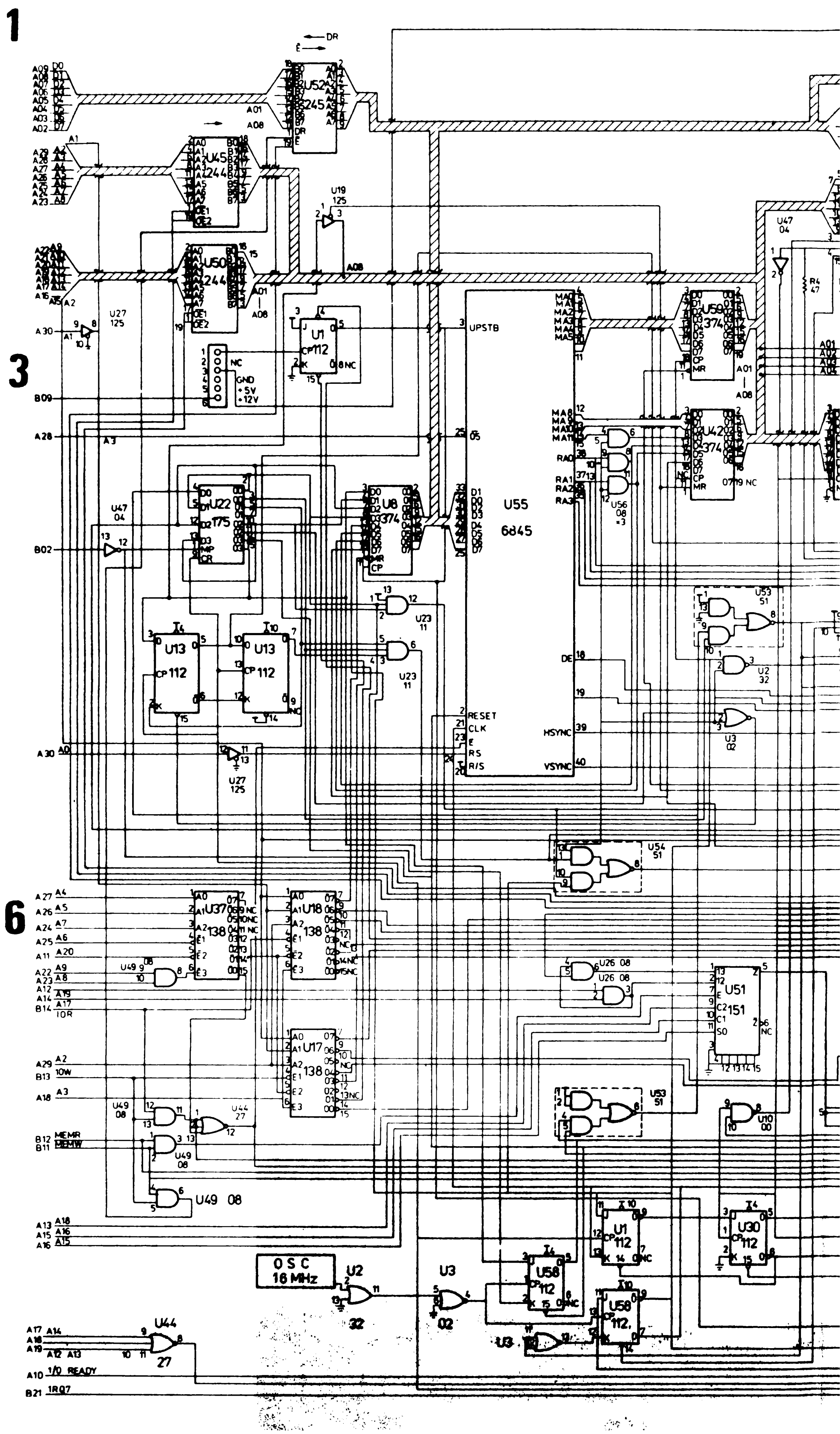
5 GENERATOR SEMNE

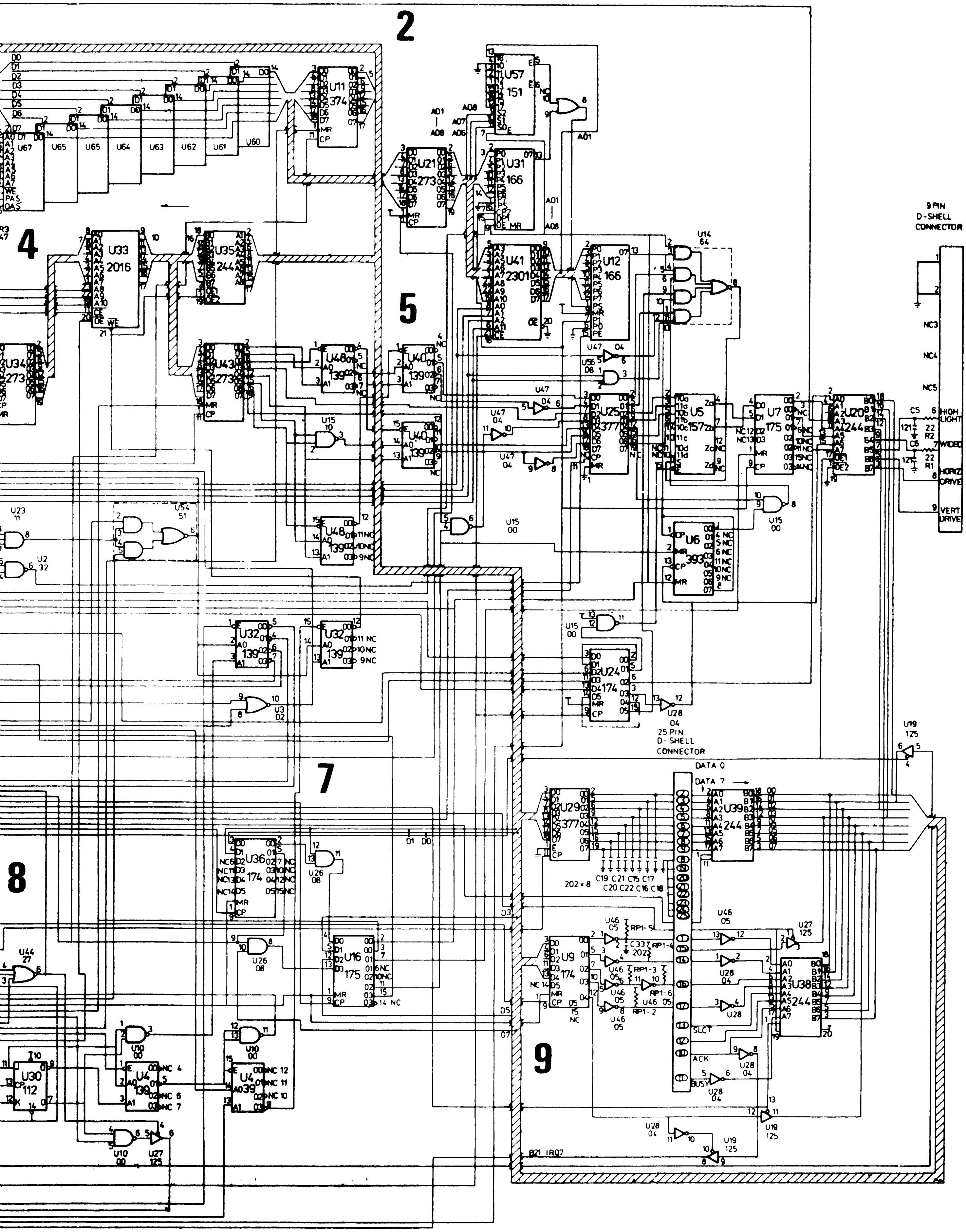
6 BLOC DECODIFICARE ADRESE PLACH.

7 REGISTRE COMANDA ILUMINARE

8 BLOC COMANDA ACCES MEMORIE IMAGINE

9 BLOC CONECTOR IMPRIMANTA





2
4
5
7
8
9
U11 374
U67
U65
U64
U63
U62
U61
U60
U33 2016
U35 244
U34 273
U43 273
U48 139
U40 139
U15
U54 51
U23 11
U2 32
U21 273
U57 151
U31 166
U41 2301
U12 166
U14 64
U47 04
U56 08
U25 377
U5 157
U7 175
U20 244
U6 393
U15 00
U32 139
U3 02
U24 174
U28 04
25 PIN D-SHELL CONNECTOR
9 PIN D-SHELL CONNECTOR
HIGH LIGHT
VIDEO
HORIZ DRIVE
VERT DRIVE
NC3
NC4
NC5
C5
C6
R1
R2
U19 125
DATA 0
DATA 7
U29 377
U39 244
U36 174
U26 08
U16 175
U9 174
U46 05
RP1-5
RP1-4
RP1-3
RP1-6
RP1-2
C33 202
202 × 8
C19 C21 C15 C17
C20 C22 C16 C18
U27 125
U38 244
SLCT
ACK
BUSY
U44 27
U30 112
U10 00
U4 139
U4 139
B21 IRQ7

Conector imprimantă

Pin	Polarizare — registru — direcție	Obs.
1	— STROBE/OUT	impuls selecție date
2-9	D0-D7/OUT	date
10	— ACK/IN	impuls confirmare primire date
11	+ BUSY/IN	echipament ocupat
12	+ END PAPER/IN	sfîrșit hîrtie
13	+ SELECT STATUS/IN	imprimare-ready
14	— AUTO FEED/OUT	comandă hîrtie
15	— ERROR/IN	eroare imprimare
16	— INIT/OUT	inițiere imprimantă
17	+ SELECT IN/OUT	activare imprimată
18-25	GROUND	masa

Registru display

(b_0, b_2, b_4, b_6 — neutilizate)

a) mod	bit	opțiune
+ SELECT GRAPHIC MODE	b_1	0 — mod text 1 — mod grafic 1 — deconectare imagine
+ ENABLE CHAR BLINK	b_5	0 — deconectare 1 — conectare pîlpîire
+ PACE NO.	b_7	0 — pagina 0 1 — pagina 1
+ ENABLE VIDEO OUTPUT	b_3	0 — întunecare ecran
b) stare		
+ HORIZONTAL SYNC	b_0	0 — semne normale 1 — sincronizare orizontală, ecran întunecat
+ VIDEO OUTPUT	b_3	0 — ecran neactiv
— VERTICAL RETRAGE	b_7	0 — retur spot, ecran întunecat 1 — afișare activă

Registru configurație

Bit	opțiune
b_0	0 — asigurare împotriva conectării mod grafic 1 — permitere conectarea mod grafic
b_1	0 — interzicere alegere pagina 1 1 — permitere alegere pagina 1

Conținutul imaginii provine din memoria DRAM (U60 - 67) de cîte 64 kb dinamică și U33 - 2kB statică, pentru 2 000 atribute.

Interfață imprimantă

Pe placheta HGC se află interfața CENTRONICS paralelă pentru imprimantă. Comanda se face prin 3 registre in-out: PRINTER DATA PORT - 3BCH, PRINTER CONTROL PORT - 03BDH.

Comanda imprimantă

(b_5, b_6, b_7 — nefolosiți)

+ STROBE	b_0	0 — impuls de scriere 1 — stingere impuls (= deconectare alimentare)
+ AUTO FEED 1	b_1	0 — avans automat hîrtie 1 — avans comandat de la distanță
— INIȚIALIZARE	b_2	0 — inițiere imprimantă 1 — funcționare normală
+ SELECT	b_3	0 — deconectare logică imprimantă 1 — conectare
	b_4	0 — mascare IRQ 7 1 — deconectare IRQ 7 (imprimanta gata de recepționat date)

Starea imprimantă

(b_0, b_1, b_2 neutilizați)

— ERROR	b_3	0 — greșeală imprimantă 1 — funcționare corectă
+ SELECT STATUS	b_4	0 — deconectare imprimantă 1 — conectare imprimantă
+ PAPER OUT	b_5	0 — funcționare normală 1 — lipsă hîrtie
— ACK	b_6	0 — imprimanta prelucrează date 1 — imprimanta poate primi date
— BUSY	b_7	0 — imprimanta ocupată (neactivă, defectă) 1 — imprimanta ready

Registri index

Index	Descriere	Mod grafic	Mod text
0	Rînd complet, inclusiv sincronizare	35 H	61 H
1	Număr semne vizibile pe ecran	3 DH	50 H
2	Plasare semn de inițializare sincro. oriz.	2 EF	52 H
3	Nr. semne în impuls sincro. oriz.	07 H	0 FH
4	Nr. total rînduri în imagine	5 BH	19 H
5	Nr. linii +4 pentru obținerea a 50 imagini/secundă	02 H	19 H
6	Număr rînduri vizibile	57 H	19 H
7	Număr rînduri început sincronizare Verticală (impuls sincro durează peste 16 linii orizontale)	57 H	19 H
8	Mod afișare (0 sau 2 imagine normală)	02 H	02 H
9	Număr linii în rînd	03 H	0 DH
10	Linia din rînd unde cursorul este peste semn	00 H	08 H
11	Linia din rînd de la care cursorul nu se mai suprapune peste semn	00 H	0 CH
12, 13	Numărul semnului de la care începe afișarea	00 H	00 H

Atenție!

Experimentarea cu parametrii imaginii fără cunoașterea exactă a descrierii tehnice a circuitului CRTC 6845 - MOTOROLA poate duce la distrugerea monitorului.

(urmare din pag.22)

```
LIB.H:
#define          COPY     0
#define          AND      1
#define          OR       2
#define          XOR      3
```

```
LIB_BAS.C
unsigned       char     replacement_rule;

drawing_mode(mode)
        unsigned        char     mode;
        {replacement_rule = (mode<<3) &0x18;
         outp(P_ctr_a,3);
         outp(P_ctr_r,replacement_rule);
        }
```

Din pacate pentru diverse pattern-uri vom constata cît de neplăcute sînt efectele aliasing-ului. Mai putem compensa cîte ceva trecînd la un pattern pe 16 biţi, dar complicarea programului nu justifică îmbunătăţirea.

Alt atribut este grosimea liniei. Grosimea constă în trasări succesive cu alterarea cu cîte un pixel a coordonatelor liniei. Această alterare se face pe baza valorilor rotunjite ale funcţiilor sin şi cos calculate pentru panta dreptei. Deoarece nu reprezintă o problemă de algoritm grafic sau cuplor EGA/VGA deosebită, vă lăsăm plăcerea să implementaţi singuri funcţia **line—width ()** în fişierul LIB-BAS.C şi să alternaţi corespunzător funcţia line ( ) din fişierul LIB-DC.C.

Un ultim atribut, dar deosebit de spectaculos este modul de trasare. Revăzînd schema funcţională a lui EGA/VGA din numărul trecut, vom observa că scrierea unui nou pixel nu se face direct, ci printr-un ALU cu patru funcţii. Deoarece cursoarele grafice şi alte funcţii lucrează cu unele moduri bine precizate, este necesară ca şi în alte cazuri anterioare, memorarea stării într-o variabilă internă bibliotecii pentru restaurarea stării cuplorului după utilizarea primitivelor menţionate. Pentru a uşura scrierea şi citirea unor programe de aplicaţii, valorile pentru comanda funcţiilor ALU vor fi stocate în header-ul bibliotecii sub forma unor constante simbolice.

Dialog cu cititorii

Ne propunem, ca în cadrul acestei rubrici sa avem un dialog deschis cu cititorii asupra unor multiple probleme cu care se confruntă informatica în ţara noastră. Pe de altă parte, tot în paginile revistei, sub antetul „COMPUTERWORLD" răspundem la o mica parte din întrebarile care ne-au fost adresate cu privire la noutăţile din domeniu folosind bibliografia IDG pe care o primim cu regularitate la redacţie.

Revenind la rubrica „SEMNAL", ne-am hotărît, cu acordul autorului, să publicăm un fragment dintr-o scrisoare pe care am primit-o la redacţie şi care ne-a creat o mare tristeţe. Autorul, Mathe Stefan, este animatorul şi conducătorul clubului „Spectrum" din Tîrgu Mureş. Cele spuse de Stefan Mathe nu necesită comentarii. Ele sînt de prisos. Noi sperăm, însă, că aceste rînduri vor fi citite şi de cei care ar trebui să...

„Pot să vă anunţ că Clubul SPECTRUM din Tîrgu Mureş există în continuare, în ciuda tuturor greutăţilor ce s-au ivit. Nu mă pot lăuda că avem o activitate foarte bogată, dar măcar odată pe săptămînă ne întîlnim, facem schimburi de programe, documentaţii şi, mai ales, de experienţă. Aceste întîlniri au în centrul lor de cele mai multe ori o prezentare organizată a unui software utilitar, urmînd discuţii libere. Unul dintre cîştigurile acestor întruniri este că elevii care vin acolo au ocazia să butoneze puţin la cele cinci calculatoare HC-85 care ne-au rămas prin bunăvoinţa primăriei municipiului, după ce o parte din întreprinderi şi-au retras calculatoarele din sala noastră. În altă zi din săptămînă desfăşurăm cursuri de iniţiere în programarea acestor calculatoare. Începînd cu acest an, iniţierea o facem doar cu adulţi, deoarece considerăm că este mai eficient aşa, ei pot transmite mai multor copii, la rîndul lor, cunoştinţele acumulate. După cum am putut aprecia, cursul nostru este mai complex decît cel susţinut de Universitatea Populară (sau cum s-o mai fi numind acum) ori de Casa Tineretului din municipiu. Nemaivorbind că celelalte cursuri se fac contra unei taxe de 500 lei. În altă zi, cursanţii pot veni să lucreze individual, fără profesor.

Consider că avem cele mai bune condiţii pe care, la nivelul actual din dotarea învăţămîntului cu calculatoare şi-l poate imagina cineva.

Avem însă două motive majore de amărăciune:

- profesorii nu doresc cu nici un chip să înveţe, deşi este mare nevoie de cadre care să predea această disciplină (care are un capitol aparte şi în manualele de clasa a VII-a şi VIII-a - nu punem acum în discuţie nivelul acestui capitol;

- nici o şcoala nu doreşte să ne asociem, să umplem sala cu calculatoare pentru a putea lucra cu un număr mai are de cursanţi odată, ei beneficiind în aceeaşi măsură în care beneficiem noi. Calculatoarele care sînt în prezent în dotarea şcolilor sînt, în multe cazuri, ţinute acasă de unii profesori."

Încheiem aici citatul. În continuare autorul ei se referă cu spirit critic la ceeă ce a găsit şi la ceea ce nu a găsit în paginile primului număr al „INFOCLUB". Nu publicăm restul din lipsă de spaţiu tipografic, dar îi mulţumim şi raspundem pe aceasta cale, tuturor celor care ne-au întrebat, că INFOCLUB se poate obţine direct de la redacţie, prin trimiterea mandatului poştal cu contravaloarea de 1x35 sau 4x35 lei (anul acesta apar 4 numere, iar de la anul lunar) pe adresa: **GHEORGHE BADEA, Piaţa „Presa Liberă" nr. 1, Bucureşti, Cod 79 781, pentru revista „INFOCLUB".**

Vom reveni în numerele viitoare cu diferite opinii privind soarta informaticii în România.

Semnal

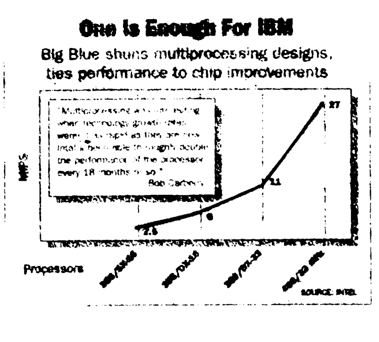
IMB lasă frîu liber... „cailor"

INFOWORLD/24 decembrie 1990

Acesta este sloganul publicitar cu care Big Blue își anunță revenirea în fruntea plutonului PC prin noile sale mașini IBM PS/2 Model 90 XP 486 și Model 95 XP 486.

Să vedem mai întîi care este componența plutonului. Pe undeva pe la coadă, mai aleargă încă mașinile XT. Pentru ele compu-gerontologia a lucrat din plin: procesoare NEC V.2x în 10 și 12 MHz, cuploare VGA cu monitoare monocrome și color, discuri fixe cu capacitate mai mare și timp de acces mai mic. Urmează apoi un grup important bazat pe microprocesorul 286 lucrînd în 14/16/20 MHz. Declinul lor este evident și marile firme păstrează în nomenclatorul lor doar unul sau două modele pentru aplicațiile nepretențioase și clienți cu bani puțini. De fapt în ultima vreme pentru aceștia au devenit mult mai accesibile și mai tentante mașinile bazate pe microprocesorul 386 SX lucrînd la 16/20 MHz. Ajungem în sfîrșit la mașina caracteristică anului 1990 care dispune de un procesor 386 lucrînd la 25/33 MHz, memorie standard de 2 MB, memorie cache internă de 32 KB, adaptor grafic Super VGA (1024x768) și discuri Winchester de cel puțin 80 MB. În aceste condiții anul 1991 va fi oare anul calculatoarelor bazate pe microprocesorul 486? Probabil că da!

În toată această evoluție nu este vorba numai de creșterea frecvenței ceasului microprocesorului și a dimensiunii bus-ului de date ci și de multe alte îmbunătățiri calitative care sporesc sensibil puterea de lucru. Iată sintetic, într-un tabel furnizat de firma Intel, evoluția procesoarelor sale:



Să revenim însă la cele două noi modele ale firmei IBM anunțate la început. Ele au o arhitectură internă sofisticată care caută să exploateze întregul potențial al microprocesorului Intel 486. Denumirea Expandable Processor (XP) evidențiază posibilitatea adăugirilor și îmbunătățirilor ulterioare care să prelungească timpul de serviciu al unui astfel de sistem într-un domeniu în care evoluțiile tehnologice se petrec cu o viteză surprinzătoare.

În ceea ce privește bus-ul intern s-a optat tot pentru Micro Channel (MCA) ca și la alți membri ai familiei PS/2 dar într-o versiune de 32 de biți și cu o viteză de transfer îmbunătățită, versiune care permite cu ajutorul cuploarelor de tip „master" să se adauge pînă la alte 16 procesoare. Deși această facilitate există, Bob Carberry — asistent al mangerului general pentru tehnologia sistemelor din departamentul de sisteme personale, a anunțat că deocamdată pentru IBM un singur procesor este de-ajuns deoarece Intel este capabilă să dubleze performanțele microprocesoarelor la fiecare 18 luni.

Cuplorul de disc este în mod surprinzător conform standardului industrial SCSI dar cu perfecționări pentru a putea suporta o viteză de transfer foarte mare. Tot în sensul creșterii ratei transferului se poate opta și pentru un cache de 256 kB! Pentru lucrul în rețea cu baze de date foarte mari există două opțiuni cu ajutorul cărora se pot gestiona resurse enorme:

	Model 90	Model 95
Hardfile Expansion Bay	0,96 GB	1,6 GB
External Storage Enclosures	8,96 GB	8,96 GB

Să nu neglijăm însă adaptorul grafic care introduce un nou standard: Extended Graphics Array (XGA). Folosind un monitor performant tip IBM 8512 avem pe lîngă modurile de lucru VGA tradiționale unele noi, deosebit de performante:

- Moduri grafice cu organizare pe plane: 640x480 — 256 culori/64 gradații de gri; 1024x768 — 16 culori/16 gradații de gri;
- Moduri grafice cu acces direct (pe 16 biți!): 640x480 — 64 K culori;
- Moduri alfanumerice: 50x132.

În final vă prezentăm un tabel cu configurațiile celor două modele:

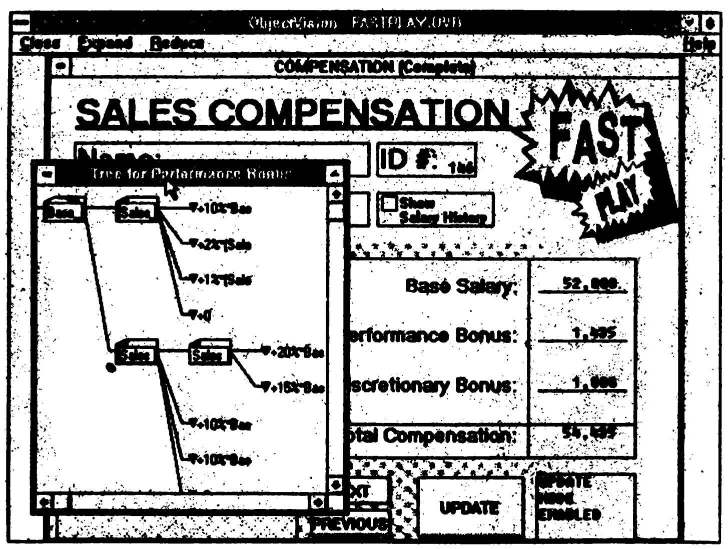
	Model 90	Model 95
MICROPROCESOR		
Standard	80486	80486
Frecvența ceas	25—33 MHz	25—33 MHz
MEMORIE		
Standard	4 MB (70 ns)	4 MB (70 ns)
Maxim	32 MB	32 MB
ADAPTOARE INTEGRATE	Extended Graphics Anay (XGA)	
	Port Serial cu DMA	
	2	1
	Port paralele cu DMA	
	controler de disc flexibil pt. 3 unități	
	controler de disc fix (SCSI) cu cache	
DISC FIX		
Standard	80—320 MB	160—320 MB
SLOT-URI LIBERE	3 x 32 biți	6 x 32 biți
BUS	MCA 32 biți	MCA 32 biți

Borland umple ferestrele goale cu Visual Apps Generator

INFOWORLD/7 ianuarie 1991

Intrarea pe piața produselor Windows anunțată cu mult timp în urmă este așteptată către mijlocul anului sub forma unui generator de aplicații care utilizează tehnicile de programare vizuală în scopul de a ajuta utilizatorii să creeze aplicații fără a utiliza un limbaj de programare.

Specialiștii de la firma Borland au confirmat că programul bazat pe Windows, ce poartă denumirea de Object Vision, permite utilizatorilor crearea aplicațiilor prin construirea unui formular pe ecran și asignarea caracteristicilor și comportamentului, fiecărui element al formularului. Aceste caracteristici sînt prezentate sub forma unui arbore decizional cu trasee care ajută utilizatorul să „navigheze" în interiorul formularului.



Formularele Object Vision pot reprezenta formulare reale, ca de exemplu formulare medicale, dar sînt, de asemenea, și o formă simbolică pentru stocarea și accesul datelor — foarte asemănător cu fișele și stivele de la Hypercard (de la Apple). Chiar dacă Object Vision are aceleași caracteristici funcționale ca și alte produse similare, programul se situează deasupra acestor capabilități prin interconectarea unor facilități de spreadsheet și gestiunea bazelor de date.

Firma din Scotts Valley, California, și-a remodelat în mod aparent produsul de la debutul liniștit la închiderea Comdex-ului, cînd el era numit Expresso și descris ca un produs pentru formularistică. Analiști familiarizați cu Object Vision au afirmat că legătura sa strînsă cu alte aplicații îl situează cu o clasă deasupra procesoarelor de formularistică (spreadsheet). „Este un sistem pentru construit utilitare", a afirmat Jeff Tarter, publicist la Softletter, o publicație de „scandal" din mediul industrial.

Object Vision, care poate accesa datele în mod concurent din surse multiple ce se află în rețea, poate lucra cu Dbase, Btrieve, Paradox, fișiere ASCII cu cîmpurile delimitate prin virgulă și aplicații care suportă schimburile dinamice de date sub Windows. Legătura cu raportările bazate pe limbajul SQL vor fi implementate mai tîrziu.

Logica lucrului cu Object Vision este foarte asemănătoare cu logica declarativă din alte produse pentru spreadsheet. Utilizatorul poate defini cîmpuri, fie în interiorul unui formular, fie în interiorul unei baze de date de o manieră similară cu celelalte din tabele. Din surse ale companiei, s-a afirmat că Borland situează Object Vision ca un superset al unui pachet spreadsheet.

Surse autorizate din mediul industrial afirmă că Borland va ridica nivelul lui Object Vision deasupra masei actuale de utilizatori. „Borland este deja în frunte pe această direcție de multă vreme și are deja un număr mare de clienți care au realizat aplicații cu Quattro, Paradox și limbajele bazate pe ele", a afirmat Tarter.

Compania subliniază, de asemenea, diferența între Object Vision și integratorul de date denumit Info Alliance de la Software Publishing Corp, despre care Borland afirmă că nu poate fi considerat un generator de aplicații pentru utilizatorul curent. De altfel, Info Alliance pornește de la 8 500 $ pentru 10 utilizatori, pe cînd Object Vision va costa „prețul mediu al unui produs spreadsheet", au afirmat oficiali ai firmei Borland.

Traducere și adaptare de Eugen GEORGESCU

Ion Diamandi,
Florin Vasiliţă

RUTINĂ GRAFICĂ pentru UMPLERA UNOR CONTURURI

Răspundem prin acest articol numeroaselor scrisori şi solicitări din partea utilizatorilor de calculatoare personale, în speranţa că, în acest fel, vom satisface nevoile mult mai multor posesori de astfel de calculatoare care, din diferite motive, nu au reuşit să ne scrie pînă în acest moment. Prezentăm deci, o rutină foarte performantă realizată în cod maşină şi utilizabilă în programe pentru umplerea rapidă a unor contururi.

Principiul de realizare a rutinei şi descriere. Se caută începutul liniei de pixeli în care se află punctul curent analizat. De aici (de la început) se umple linia pînă la găsirea unui punct activat sau a sfîrşitului (marginei) de ecran. Dacă se găseşte o ramificaţie, aceasta este depusă în lista FIFO. Dacă în listă se găseşte o ramificaţie, atunci se aplică umplerea de linie şi se scoate din listă. Noua ramificaţie se depune la sfîrşitul listei iar citirea se face de la început.

În locul coordonatelor punctelor se utilizează adresa + mască. În mască un bit setat va da un punct. Lista FIFO are 400 de locaţii (o locaţie = 3 octeţi). WRITER indică următoarea locaţie liberă, iar READER, ramificaţia cea mai veche.

PUTTUB depune o ramificaţie, iar GETTUB extrage o ramificaţie. Umplerea unui rînd se realizează cu subrutina PLINE. Prin intermediul subrutinelor HLDOWN şi HLUP se solicită o adresă din memoria RAM pentru ecran şi se furnizează adresa octetului care este „sub" şi, respectiv, „deasupra" adresei. Dacă lista FIFO este plină PUTTUB returnează Z = 1.

Performanţe: rutina poate umple pînă la maximum 10 000 puncte (pixeli) pe secundă (de exemplu, un patrulater de 100 x 100 este umplut în circa o secundă).

Rutina de umplere în cod maşină pentru calculatoare HC:

```
        ORG   #F000

PAINT:
        PUSH  AF
        PUSH  BC
        PUSH  DE
        PUSH  HL
        CALL  TUBINI
        CALL  PIXADD
        CALL  MASKG
        CALL  MODE
        CALL  PUTTUB

PAINT1:
        CALL  PLINE
        JR    NZ.PAINT1
        POP   HL
        POP   DE
        POP   BC
        POP   AF
        RET

MODE:
        LD    C.0
        CALL  POINT
        RES   3.C
        RET   Z
        SET   3.C
        RET

TUBINI:
        LD    HL.FIFO
        LD    (READER).HL
        LD    (WRITER).HL
        LD    DE.(WRITER)
        RET

GETTUB:
        LD    HL.(READER)
        LD    DE.(WRITER)
        CALL  CP16
        RET   Z
        INC   HL
        LD    E.(HL)
        INC   HL
        LD    D.(HL)
        INC   HL
        LD    B.(HL)
        PUSH  DE
        LD    DE.FIFOE
        CALL  CP16
        JR    NZ.GET1
        LD    HL.FIFO

GET1:
        LD    (READER).HL
        POP   HL
        XOR   A
        INC   A          ;NZ!
        RET

PUTTUB:
        EX    DE.HL
        PUSH  DE
        LD    HL.(WRITER)
        INC   HL
        LD    (HL).E
        INC   HL
        LD    (HL).D
        INC   HL
        LD    (HL).B
        LD    DE.FIFOE
        CALL  CP16
        JR    NZ.PUT1
        LD    HL.FIFO

PUT1:
        LD    DE.(READER)
        CALL  CP16
        JR    Z.PUT2
        LD    (WRITER).HL

PUT2:
        POP   HL
        RET

PLINE:
        CALL  GETTUB
        RET   Z
        CALL  SLINE
        SET   0.C
        SET   1.C

PLINE1:
        LD    A.(HL)
        CALL  POINT8
        JR    Z.PLINE2

PLINE3:
        CALL  POINT
        RET   NZ
        CALL  PLOT
        CALL  BOT1
        CALL  TOP1
        RRC   B
        JR    NC.PLINE3
        JR    PLINE5

PLINE2:
        XOR   A
        BIT   3.C
        JR    NZ.PLINE4
```

```
        CPL
PLINE4:
        LD   (HL).A
        CALL TOP8
        CALL BOT8
PLINE5:
        INC  HL
        LD   A.L
        AND  #1F
        JR   NZ.PLINE1
        INC  A          ;NZ!
        RET
POINT:
        LD   A.(HL)
        AND  B
        BIT  3.C
        JR   NZ.PO1
        OR   A
        RET
PO1:
        XOR  B
        RET
PLOT:
        LD   A.(HL)
        OR   B
        BIT  3.C
        JR   Z.PL1
        XOR  B
PL1:
        LD   (HL).A
        RET
BOT1:
        PUSH HL
        CALL HLDOWN
        LD   DE.SCREND
        CALL CP16
        JR   NC.BOT12
        CALL BOTB
BOT12:
        POP  HL
        RET
BOTB:
        CALL POINT
        JR   Z.BOTB1
        SET  0.C
        RET
BOTB1:
        BIT  0.C
        RET  Z
        CALL PUTTUB
        RES  0.C
        RET
TOP1:
        PUSH HL
        CALL HLVP
        LD   DE.SCREEN
        CALL CP16
        JR   C.TOP12
        CALL TOPB
TOP12:
        POP  HL
        RET
TOPB:
        CALL POINT
        JR   Z.TOPB1
        SET  1.C
        RET
TOPB1:
        BIT  1.C
        RET  Z
        CALL PUTTUB
        RES  1.C
        RET
HLDOWN:
        INC  H
        LD   A.H
        AND  7
        RET  NZ
        PUSH DE
        LD   DE.-#7E0
        ADD  HL.DE
        POP  DE
        LD   A.H
        AND  7
        RET  Z
        LD   A.H
        ADD  A.7
        LD   H.A
        RET
HLVP:
        LD   A.H
        AND  7
        JR   Z.HLVP1
        DEC  H
        RET
HLVP1:
        LD   A.L
        AND  #E0
        JR   Z.HLVP2
        PUSH DE
        LD   DE.#6E0
        ADD  HL,DE
        POP  DE
        RET
HLVP2:
        LD   A.L
        SUB  #20
        LD   L.A
        DEC  H
        RET
SLINE:
        CALL LEFTB
        JR   C.SLINE1
        RRC  B
        RET
SLINE1:
        RRC  B
SLINE2:
        LD   A.L
        AND  #1F
        RET  Z
        DEC  HL
        LD   A.(HL)
        CALL POINT8
        JR   Z.SLINE2
        RLC  B
        CALL LEFTB1
        RRC  B
        RET  NC
        INC  HL
        RET
LEFTB:
        RLC  B
        RET  C
LEFTB1:
        CALL POINT
        JR   Z.LEFTB
        OR   A
        RET
CP16:
        LD   A.H   ;HL.DE
        CP   D
        RET  NZ
        LD   A.L
        CP   E
        RET
BOT8:
        PUSH HL
        CALL HLDOWN
        LD   DE.SCREND
        CALL CP16
        JR   NC.BOT8N
        LD   A.(HL)
        OR   A
        JR   Z.BOT81
        CPL
        JR   NZ.BOT82
BOT81:
        CALL BOTB
        POP  HL
        RET
BOT82:
        CALL BOTB
        RRC  B
        JR   NC.BOT82
BOT8N:
        POP  HL
        RET
TOP8:
        PUSH HL
        CALL HLVP
        LD   DE.SCREEN
        CALL CP16
        JR   C.TOP8N
        LD   A.(HL)
        OR   A
        JR   Z.TOP81
        CPL
        JR   NZ.TOP82
TOP81:
        CALL TOPB
        POP  HL
        RET
TOP82:
        CALL TOPB
        RRC  B
        JP   NC.TOP82
TOP8N:
        POP  HL
        RET
POINT8:
        BIT  3.C
        JR   Z.PO81
        CPL
PO81:
        OR   A
        RET
MASKG:
        LD   B.1
        INC  A
MG1:
        RRC  B
        DEC  A
        JR   NZ.MG1
        RET
FIFO    EQU  #
        DEFS 400*3
FIFOE   EQU  #-1
READER  DEFS 2
WRITER  DEFS 2
PIXADD EQU   #22AA ;
 RUTINA  EPROM
SCREEN EQU   #4000
SCREND EQU   #57FF
TEST    LD   BC.(#5C7D) ;
ULTIMUL PUNCT
        JP   PAINT
        END  TEST
```

(continuare în pag.47)


UTILIZATORULUI
Thinking about your help
A B C

Stelian NICULESCU
Cristian ARTEMI

SISTEMUL DE OPERARE MS-DOS PENTRU CALCULATOARELE PERSONALE

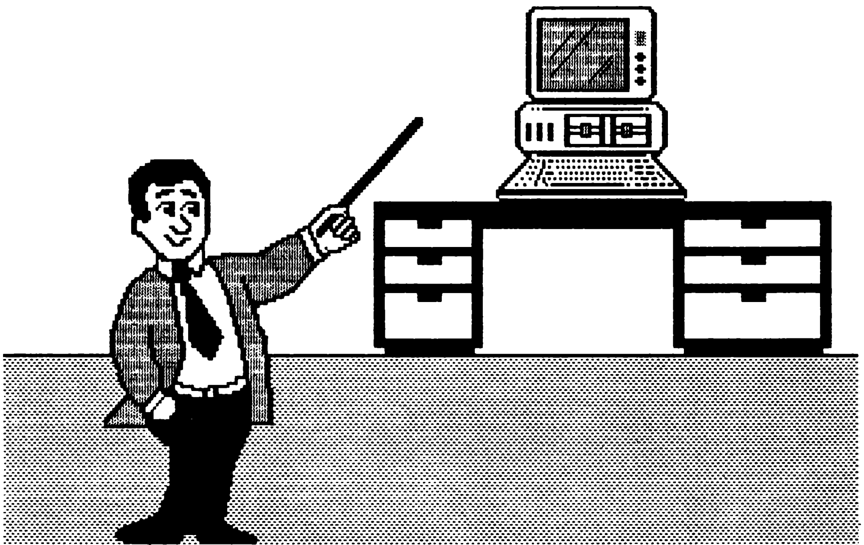
Introducere

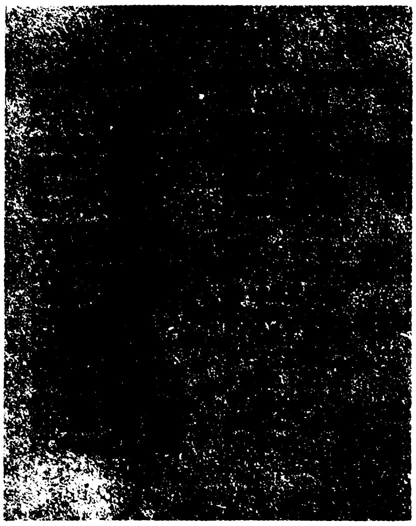
Un sistem de operare este un ansamblu de programe care realizează legătura, comunicarea, dintre hardware, utilizatorul uman şi alte sisteme software.

Sistemele de operare sînt strîns legate de maşină, de aceea ele sînt specifice fiecărui tip de calculator. Pentru cele bazate pe microprocesoarele 8086 şi 8088 (IBM-PC compatibile), firma americană Microsoft Corporation a realizat sistemul de operare MS-DOS (Microsoft-Disk Operating Sistem) ajuns la versiunea 4. 0. Acesta este păstrat pe un suport de memorie externă numit disc sistem. Utilizatorul îşi poate copia sistemul pe orice disc flexibil sau pe un disc rigid (dacă calculatorul îl conţine în configuraţie). (El este specific familiei de procesoare pe 16 biţi 8086/8088 şi nu poate fi utilizat pentru alte tipuri ca Z 8000 de exemplu). MS-DOS asigură comunicarea cu calculatorul, unităţile de discuri, imprimanta, mouse-ul şi alte periferice, gestionînd aceste resurse.

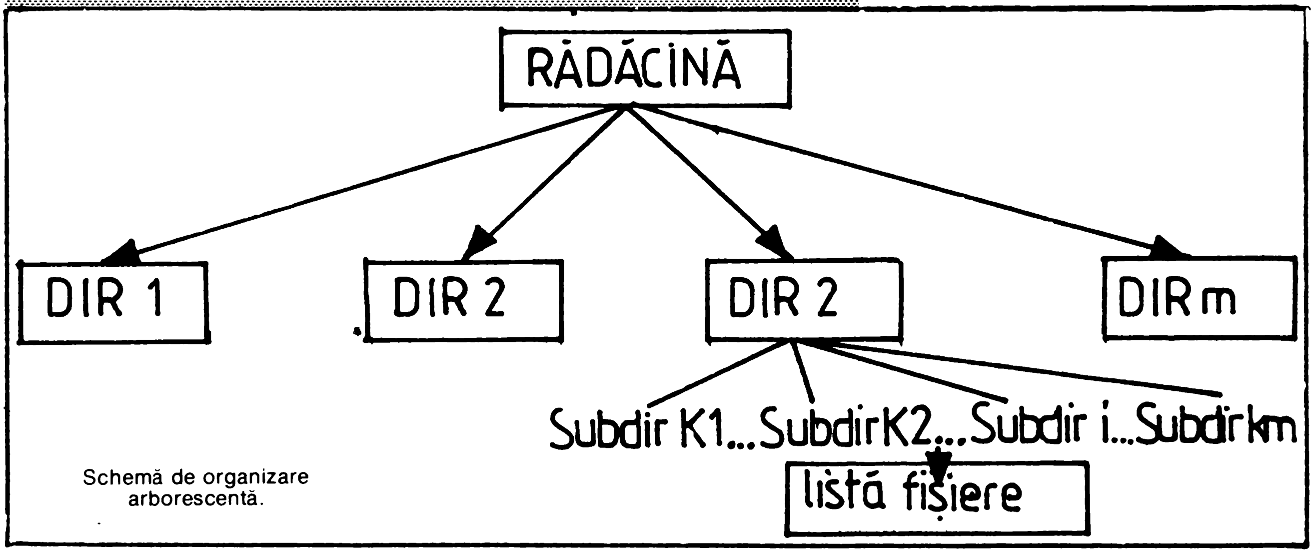
El dă posibilitatea utilizatorului: să creeze şi să prelucreze fişiere; să înlănţuiască şi să execute programe; să acceseze echipamente periferice ataşate calculatorului.

Cea mai mare parte a informaţiilor (date şi instrucţiuni) memorate pe suporturi externe sînt organizate în fişiere. Un fişier este o colecţie de informaţii de acelaşi tip. Discul sistem conţine următoarele fişiere MS-DOS (care conţin comenzi ale sistemului de operare sau informaţii necesare acestuia):





MS-DOS permite organizarea fişierelor pe discuri în directoare (moduri de împărţire a fişierelor în grupuri, în funcţie de opţiunea utilizatorului). Un director poate conţine un număr de fişiere dar şi alte directoare numite subdirectoare. Astfel se poate obţine o structură ierarhică de directoare de tip arborescent care se poate mări, prin crearea unor noi directoare (respectiv subdirectoare).



Schemă de organizare arborescentă.

Discul pe care se lucrează se numeşte curent, iar directorul în care se lucrează, director curent. Dacă se face o referire la un fişier memorat pe un disc sau într-un alt director, decît cel curent, acestea trebuie specificate prin:

— identificatorul (litera) de disc urmat de simbolul „:" şi

— drumul, din arborele director al discului, prin care se ajunge la fişier. Exemplu: A:/DIR 2/SUBDIR 3/ nume fişier extensie, unde nume fişier este format din cel mult 8 caractere (litere, cifre, „—"), iar extensia din cel mult 3 caractere, alese de utilizator.

Directoarele conţin şi informaţii referitoare la dimensiunea fişierelor, poziţia lor pe disc şi datele cînd au fost create sau actualizate. O zonă adiţională a sistemului, creată pe fiecare disc, este tabela de alocare fişiere (File Allocation Table). In ea sînt localizate fişierele pe disc; astfel pot fi alocate spaţii libere pentru a crea noi fişiere. Aceste două zone sistem, directoarele şi tabela de alocare, ajută ca MS-DOS să recunoască şi să organizeze fişiere pe discuri. La iniţializarea unui disc, MS-DOS crează tabela de alocare şi un director gol numit directorul rădăcină.

Încărcarea sistemului

Pentru iniţializare se introduce discul sistem în unitatea de disc logic A (în manualul calculatorului se specifică care este aceasta) şi se deschide monitorul şi calculatorul. Discul va fi citit şi anumite fişiere (nucleul sistemului) care conţin informaţii necesare permanent sînt introduse în memoria calculatorului. Pe ecran apare mesajul:

Current date is True 1-01-1980
Enter name date (mm - dd - yy):

MS-DOS aşteaptă de la utilizator data scrisă după formatul din paranteză. Exemplu: Data de 7 Martie 1990 este descrisă prin 07-03-90, după vare se apază tasta ENTER (sau RETURN la unele calculatoare). Precizăm că notaţia «nume» specifică acţiunea apăsării tastei nume.

Pe ecran va apare un nou mesaj:

Current time is 0: 00:45:10
Enter new time:

şi se aşteaptă introducerea timpului.Exemplu: Ora 13 şi 30 minute este descrisă prin 13:30 «ENTER»

Următorul mesaj este:

Microsoft (R) MS-DOS (R) Version 3.30
(c) Copyright Microsoft Corp 1981-1987
A»-

In acest moment discul curent este A şi se aşteaptă o comandă de la utilizator. Dacă se doreşte schimbarea unităţii curente de disc se tastează identificatorul său urmat de „:". Exemplu: A»B: «ENTER» şi pe ecran apare B»- şi deci discul B devine disc curent.

Comenzi MS-DOS

Comenzile sînt căi de comunicare cu calculatorul. O comandă MS-DOS se introduce la terminal pe o linie numită linie de comandă. Prin apăsarea tastei ENTER comanda este automat trimisă la procesorul de comandă (COMMAND.COM) pentru execuţie. În acelaşi timp o copie este trimisă către o memorie numită linie tampon de unde comanda poate fi rechemată şi modificată prin utilizarea unor taste speciale de editare MS-DOS.

Comenzile MS-DOS realizează următoarele funcţii:

- Comparaţie, copiere, afişare, ştergere şi redenumire fişiere;
- Copiere şi formatare (iniţializare) discuri;
- Executare programe de sistem;
- Analiză şi listare directoare;
- Introducere dată, timp şi comentarii;
- Iniţializare opţiuni de imprimantă sau ecran;
- Copiere fişiere sistem MS-DOS pe un alt disc;
- Cererea către MS-DOS de a aştepta o perioadă anume de timp.

Comenzile sistemului sînt de două tipuri: interne şi externe. Comenzile interne sînt cele mai simple şi mai des utilizate, fiind părţi ale procesorului de comandă care este încărcat în memorie. De aceea, cînd sînt tastate, ele se execută imediat. În continuare prezentăm o listă a acestora (în paranteză sînt denumiri echivalente):

BREAK	**DEL (ERASE)**	**MKDIR (MD)**	**SET**
CHDIR(CD)	**DIR**	**PATH**	**SHIFT**
CLS	**ECHO**	**PAUSE**	**TIME**
COPY	**EXIT**	**PROMPT**	**TYPE**
CTTY	**FOR**	**REM**	**VER**
DATE	**GOTO**	**REN(RENAME)**	**VERIFY**
	IF	**RMDIR(RD)**	**VOL**

Comenzile externe se află pe disc ca fişiere program. După tastarea unei comenzi de către utilizator, calculatorul trebuie să o citească. Dacă nu o găseşte pe disc, MS-DOS nu poate executa comanda şi semnalează printr-un mesaj de eroare. Orice fişier de extensii COM, EXE sau BAT poate fi considerat o comandă externă.

Observaţie: introducerea unei comenzi externe nu trebuie să includă şi extensia de fişier. Lista comenzilor externe regăsite pe discul sistem este:

ASSIGN	**COMP**	**FIND**	**LABEL**	**PRINT**	**SORT**
ATTRIB	**DISKCOMP**	**FORMAT**	**MODE**	**RECOVER**	**SUBST**
BACKUP	**DISKCOPY**	**GRAPHICS**	**MORE**	**RESTORE**	**SYS**
CHKDSK	**EXE2BIN**	**JOIN**	**PARK**	**SHARE**	**TREE**

Valeriu IORGA

UTILIZAREA SISTEMULUI TURBO PASCAL VERSIUNILE 5.0 şi 5.5

1. Fişiere necesare utilizării versiunilor 5.0 sau 5.5.

În versiunile 5.0 şi 5.5. utilizatorul are la dispoziţie două sisteme TURBO Pascal şi anume:
— un *mediu integrat de dezvoltare* (TURBO.EXE), care prin intermediul unor meniuri şi ferestre permite editarea, compilarea, execuţia şi depanarea programului;
— un compilator independent (TPC.EXE), ocupînd numai 225 kocteţi de memorie, numit *compilator în linie de comandă.*

Pentru utilizarea deplină a tuturor facilităţilor acestor versiuni mai sînt necesare următoarele fişiere:
— biblioteca de unităţi standard TURBO.TPL, care cuprinde unităţile System, Crt, Dos, Overlay şi Printer.
— unitatea grafică GRAPH.TPU;
— unităţile de compatibilizare cu versiunea 3.0: GRAPH3.TPU şi TURBO3.TPU;
— textul help inclus în mediul de dezvoltare (TURBO.HLP);
— driverele dispozitivelor grafice • .BGI;
— bibliotecarul TPUMOVER.EXE, permiţînd adăugarea şi scoaterea de unităţi din biblioteca de unităţi standard TURBO.TPL;
— utilitarul TCONFIG.EXE pentru conversia între fişierele de configurare TURBO.TP şi TPC.CFG utilizate de către cele două compilatoare;
— utilitarul TINST.EXE pentru instalarea mediului integrat de dezvoltare;
— utilitarul BINOBJ.EXE pentru conversia fişierelor din format binar în format .OBJ;

2. Instalarea mediului integrat de dezvoltare TURBO Pascal.

Instalarea TURBO Pascal-ului pe dischete sau disc dur Winchester presupune despachetarea fişierelor corespunzătoare (din formatul compact .ARC) şi gruparea lor în subdirectoare, operaţie care se face uzual cu utilitarul INSTALL; în lipsa acestuia se copiază fişierele, care au fost în prealabil decompactate cu UNPACK. Pentru funcţionarea corectă a sistemului TURBO Pascal se recomandă ca în fişierul de configurare CONFIG.SYS să se facă asigurările: FILES=20 şi BUFERS=20.

Mediul integrat de dezvoltare poate fi folosit fără a-i face nici o modificare copiind fişierul TURBO.EXE de pe discul de distribuţie.

Adaptarea mediului integrat de dezvoltare TURBO Pascal se face cu utilitarul TINST, care permite următoarele:
— stabilirea unor căi de acces la subdirectoarele în care sînt plasate fişierele incluse, fişierele de configurare, unităţile şi fişierele executabile;
— personalizarea editorului de text TURBO Pascal;
— modificarea opţiunilor implicite ale compilatorului, editorului de legături şi debuggerului;
— modificarea culorilor;
— schimbarea dimensiunilor ferestrelor.

Pot fi generate mai multe copii de sisteme de dezvoltare cu diferiţi parametri. Programul de instalare se lansează prin:
TINST [cale\] [/B]

Copia adaptată a fişierului TURBO.EXE, avînd numele specificat va fi plasată în subdirectorul precizat prin calea de acces. Parametrul /B forţează afişarea monocromă. TINST este comandat prin meniuri. Ieşirea din TINST se face prin acţionarea tastei «ESC» din meniul principal de instalare; acţionarea aceloraşi taste dintr-un submeniu face revenirea la meniul ierarhic superior.

Selectarea unei comenzi dintr-un meniu se face poziţionînd, prin intermediul tastelor cu săgeţi o linie cursor pe comandă dorită şi acţionînd apoi ⟨ENTER⟩ sau tastînd litera corespunzătoare comenzii alese.

Meniul principal de instalare dispune de următoarele comenzi (submeniuri):
— *Compile* permite specificarea numelui implicit al Fişierului Primar compilat şi a destinaţiei compilării (în Memoria sau pe Disc);
— *Options* permite setarea unor valori implicite pentru opţiunile de compilare (verificarea domeniului, verificarea depăşirii stivei, verificarea erorilor de intrare/ieşire, alinierea datelor la limita de cuvînt, evaluarea optimizată a expresiilor booleene, procesarea numerică prin software, emularea coprocesorului, generarea de informaţii pentru debugger, dimensionarea stivei);
— *Linker* pentru setarea unor opţiuni implicite ale Editorului de Legături;
— *Environment* posedă subcomenzi pentru salvarea automată a fişierului de configurare la folosirea comenzilor **Run, File OS Shell** sau **File/Quit**, salvarea fişierului editat la folosirea aceloraşi comenzi, păstrarea penultimei versiuni a fişierului editat — fişierul. BAK, suprimarea efectului de lupă pentru fereastra activă, care nu va mai ocupa întreg ecranul, setarea unor opţiuni implicite pentru editorul de text;
— *Directory* permite specificarea căilor la subdirectoarele în care se află fişierul de configurare, fişierul help şi fişierele unităţi.
— *Debug* fixează opţiunile implicite ale debuggerului încorporat în mediul integrat de dezvoltare;
— *Editor* permite redefinirea comenzilor editorului;
— *Mode for Display* asigură folosirea unuia din modurile: Color, alb-negru, monocrom, LCD;
— *Set Colours* permite modificarea culorilor;
— *Resize Windows* modifică dimensiunile ferestrelor Edit şi Output Watc;
— *Quit/Save* permite ca modificările operate să fie instalate permanent în TURBO Pascal;

3. Utilizarea mediului integrat de dezvoltare TURBO Pascal

Lansarea sistemului TURBO Pascal se poate face folosind dischetă sau disc dur Winchester cu comanda:
TURBO
sau forma mai generală:
TURBO [fişier—sursă] [/C fişier configurare] [/B] [/D] [/M] [/P]
În acest din urmă caz:
— fişierul sursă indicat va fi încărcat în editor;
— parametrii de configurare vor fi extraşi din fişierul de configurare precizat (implicit aceştia sînt preluaţi din TURBO.TP);
— /B declanşează opţiunea Build — de recompilare a sursei, fişierelor incluse şi unităţilor folosite;
— /M declanşează opţiunea Make — de recompilare selectivă după dată a sursei, fişierelor incluse şi unităţilor folosite;

Aceste două opţiuni conduc la ieşirea din mediul integrat de dezvoltare.
— /D permite funcţionarea cu două adaptoare grafice conectate la două monitoare: un ecran primar pentru ieşirile programului şi un ecran secundar pentru mediul integrat TURBO. Comutarea între ele se face prin ⟨Alt⟩—⟨F5⟩;
— /P memorează paleta de culori curentă la comutarea ecranului;
TURBO Pascal este comandat de meniuri şi parametri necesari la compilare, fixaţi prin intermediul acestor meniuri.

Valorile parametrilor astfel modificați vor înlocui în fișierul de configurare valorile fixate la instalare cu TINST.
Meniul principal are forma:

(tabelul 1)

O comandă din meniul principal se alege folosind inițiala numelui ei sau linia cursor, mutată cu tastele săgeți și după care se acționează <**Enter**>.
Toate comenzile din meniul principal (cu excepția comenzii Edit) deschid alte submeniuri cu subcomenzi specifice, care se selectează cu o linie cursor poziționată cu ↑ sau ↓ și acționare <**Enter**>;
Dintr-un submeniu se poate ieși:
— în submeniul ierarhic superior cu <**Esc**>;
— în meniul principal cu <**Alt**> <**litera**> **(litera fiind F, E, R, C, O, D, B)**
— în DOS cu <**Alt**> <**X**>;
Pe ecran apar trei ferestre: Edit, Watch și Output dintre care una va fi activă.
Fereastra Edit devine activă:
— la selectare E în linia cursor și la acționare <**Alt**> <**E**> din alt meniu.
Acționare <**Esc**> din meniul principal conduce la precedenta fereastră activă;
<**F6**> servește pentru comutarea între fereastra Edit și Output/Watch, în timp ce comutarea între acestea se face cu <**Alt**> <**F6**>. (fig. 1)

Meniul FILE — gestiune fișiere

Servește pentru încărcarea și salvarea programelor, înregistrarea programelor sub alt nume, declanșare comenzi DOS (fig. 2).

Load. (F3) Încarcă un program în editor. În fereastra care se deschide se indică numele programului. Se pot folosi și caractere, generice (• și ?) afișîndu-se lista fișierelor corespunzătoare. Pentru selectarea unui subdirector se introduce <**Shift**> și prima literă a numelui acestuia.
Pick (Alt-F3). Încarcă în editor fișierul specificat din lista celor mai recent folosite 8 fișiere (Lista Pick).
New. Șterge memoria de lucru, pregătind editarea unui nou program.
Save. (F2) Salvează pe disc programul în curs de editare sub numele dat fișierului în momentul încărcării.
Write to. Salvează programul în curs de editare sub un nou nume.
Director. Afișează directorul fișierelor. Se pot folosi caractere generice pentru a selecta fișiere. Pot fi listate și subdirectoare.
Change dir. Afișează directorul curent și permite modificarea unității și subdirectorului selectat.
OS Shell. Apelează interpretorul de comenzi COMMAND.COM, permițînd executarea unei comenzi Dos. Revenirea în TURBO Pascal se face cu EXIT.
Quit. (Alt-X). Termină sesiunea TURBO Pascal.

Meniul RUN — execuție program.

Comandă execuția unui program de sine stătător sau împreună cu debuggerul, după care se revine în meniul principal (fig. 3).

Run (Ctrl-F9). Apelează comanda MAKE înaintea execuției programului.
Program reset (Ctrl-F2). Întrerupe funcționarea debuggerului și eliberează memoria ocupată de acesta.

```
File  Edit  Run  Compile  Options  Debug   Break/Watch
Line 1   Col 1   Insert   Indent  Unindent  A:NONAME.PAS
---------------------------Watch---------------------------
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F-10-Menu
```

tabelul 1

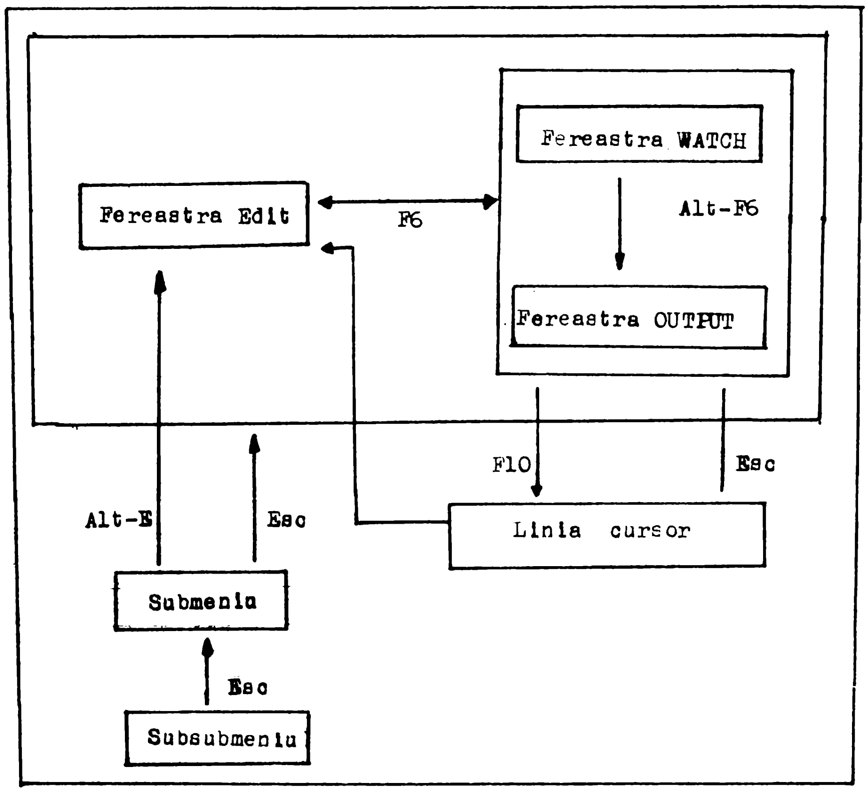


figura 1

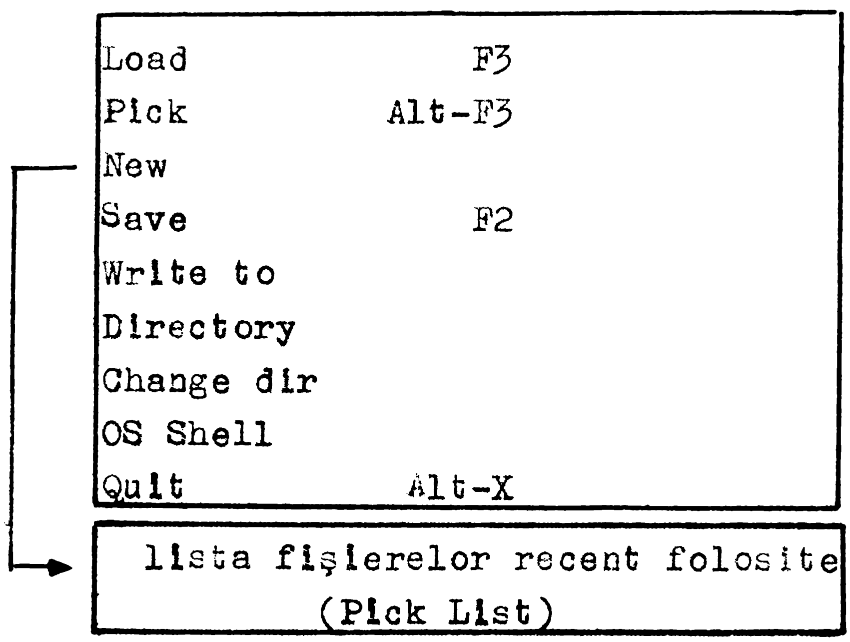


figura 2

```
Run               Ctrl-F9
Program reset     Ctrl-F2
Go to cursor           F4
Trace into             F7
Step over              F8
User screen        Alt-F5
```

figura 3

Go to cursor (F4). Execută programul pînă în linia în care se află cursorul.
Trace into (F7). Execută programul pas cu pas. Procedurile și funcțiile apelate sînt de asemenea parcurse pas cu pas.
Step over (F8). Execută programul pas cu pas. Procedurile și funcțiile apelate sînt executate „dintr-o dată".
User screen (Alt-F5). Afișează ecranul de execuție (cu rezultatele programului).

Meniul COMPILE — compilare program. (fig. 4)

```
Compile           Alt-F9
Make                  F9
Build
Destination       Memory
Find error
Primary file
Get info
```

figura 4

Compile (Alt-F9). Compilează programul în curs de editare; nu apelează utilitarele MAKE sau BUILD.
Make (F9). Apelează utilitarul MAKE: toate fișierele OBJ și INCLUDE sînt testate privind data ultimei actualizări — cele care au fost modificate între timp vor fi recompilate.
Build. Apelează utilitarul BUILD: acesta recompilează toate fișierele UNIT, OBJ și INCLUDE.
Destination. Precizează unde se depune rezultatul compilării: în Memorie sau pe Disc într-un fișier .EXE.
Find error. La producerea unei erori la execuție se indică codul erorii și adresa acesteia. Această informație servește pentru localizarea erorii în codul sursă.
Primary file. Specifică fișierul .PAS care va fi compilat, independent de fișierul care a fost editat.
Get info. Dă informații asupra programului în curs: mărimea fișierlor sursă, numărul de linii, dimensiunea stivei, heapului, etc.

Meniul OPTIUNI. — fixarea parametrilor de compilare (fig. 5)

Compiler. Această opțiune permite parametrizarea compilării. Parametrii se dau sub forma unor opțiuni de activare/inhibare (ON/OFF) sau a unor moduri. Toate aceste opțiuni corespund unor directive de compilare care pot fi fixate global prin meniuri sau local în fiecare program. (fig. 6)

```
Compiler
Linker
Environment
Directories
Parameters
Save options
Retrieve options
```

figura 5

Opțiunea din meniu	Valoarea implicită	Directiva comp.
Range checking	Off	$R
Stack checking	On	$S
I/O checking	On	$I
Force far calls	Off	$F
Align data	Word	$A
Overlays allowed	Off	$O
Var-string checking	Strict	$V
Boolean evaluation	Short Circuit	$B
Numeric processing	Software	$N
Emulation	On	$E
Debug information	On	$D
Local symbols	On	$L
Conditional defines		
Memory sizes		$M

figura 6

Range checking (off). Activează sau inhibă verificarea domeniilor: încadrarea indicilor tablourilor în limite, a tipurilor ordinale în subdomenii;
Stack checking (on). Cînd opțiunea este activă, compilatorul generează cod de verificare a încadrării variabilelor locale în spațiul rezervat în stivă;
I/O checking. (on). Cînd opțiunea este activă, o eroare într-o operație de intrare/ieșire produce o eroare la execuție.
Force far calls (off). Cînd opțiunea este inhibată sînt permise numai apeluri scurte (în același segment).
Align data (word). Variabilele și constantele cu tip sînt aliniate la limite de cuvînt.
Overlay allowed (off). Inhibă generarea de cod pentru segmentare; unitățile de program nu pot fi suprapuse.
Var string checking (strict). Verifică concordanța tipurilor string parametri formali și actuali ai procedurilor și funcțiilor.
Boolean evalution (Short circuit). Generează cod pentru evaluarea optimizată a expresiilor booleene.
Numeric processing (Software). Permite numai tipul real pe 6 octeți. Tipuile reale single, double, extended și comp sînt permise numai în prezența coprocesorului 8087.
Emulation (On). Activează/inhibă legătura cu biblioteca run-time care emulează coprocesorul numeric 8087, dacă acesta lipsește.
Debug information (On). activează generarea de informație pentru depanare:
Local symbols (On). Activează/inhibă generarea numelor și tipurilor variabilelor și constantelor loccale dintr-un modul.
Conditional defines. Definește simbolurile care pot fi referite în directivele de compilare condiționată.
Memory sizes. Permite specificarea explicită a necesarului de memorie pentru program și anume dimensiunea stivei (pînă la 65535 octeți, implicit se iau 16384 octeți) și limitele inferioară și superioară ale memoriei dinamice — heapul (maxim 655360 octeți).
Linker. Această opțiune comandă funcționarea editorului de legături.
Map file (Off). Specifică dacă se dorește generarea unui fișier MAP care conține informații asupra segmentelor, simbolilor publici și adreselor lor și a punctului de intrare.
Link buffer. Indică editorului de legături să folosească memoria sau discul pentru păstrarea temporară a informațiilor necesare.
Environment. Comandă prin parametri de tip ON/OFF funcționarea mediului integrat de dezvoltare.
Edit auto save ON — programul în curs de editare este salvat în mod automat înaintea fiecărei comenzi Run.
OFF — programul editat nu este salvat automat, astfel că în cazul unei erori la execuție se pierde.
Configuration auto save ON — în fișierul de configurare se înregistrează în mod automat modificările aduse parametrilor.
Backup files On sistemul păstrează o copie de siguranță a ultimei versiuni sursă editate (fișierul .BAK).
Tab size. Indică numărul de coloane care separă două tabstopuri succesive. Sînt permise valori între 2 și 16.
Zoom windows. Activarea acestei opțiuni (ON) permite mărirea ferestrei de editare EDIT și a ferestrei rezultatelor OUTPUT la dimensiunea întregului ecran.

Screen size. Permite alegerea numărului de linii pentru ecranele EGA şi VGA.

Directories. Indică subdirectoarele în care se face căutarea şi salvarea fişierelor utilizate.

Turbo directory. Precizează numele subdirectorului care conţine fişierele mediului integrat de dezvoltare.

EXE & TPU director. Precizează numele subdirectorului care conţine fişierele EXE, TPU şi MAP.

Include directories. precizează numele subdirectorului care conţine fişierele INCLUDE.

Unit directories numele subdirectorului care conţine fişierele unităţi.

Object directories — numele subdirectorului care conţine fişierele OBJ incorporate prin directiva de compilare (SL fişier).

Pick file name — permite precizarea unei liste cu numele celor mai recent activate fişiere (lista Pick).

Save options/Retrieve options. Permite refacerea sau salvarea unei parametrizări date.

Meniul DEBUG

Comandă funcţionarea depanatorului integrat şi gestionează accesul la variabile (fig. 6).

```
Evaluate                 Ctrl-F4
Call stack               Ctrl-F3
Find procedure
Integrated debugging     On
Stand-alone debugging    Off
Display swapping         Smart
Refresh display
```

figura 7

Evaluate — afişează un cadru de dialog cu trei zone pentru afişarea şi manipularea variabilelor. *Evaluate* permite indicarea unei variabile sau expresii a cărei valoare va fi indicată în zona rezultatelor (Output).

Result. Afişează valoarea expresiei sau conţinutul variabilei selectate prin Evaluate.

New value. Permite indicarea unei noi valori în zona Evaluate.

Call stack. Permite încărcarea în editor a unei proceduri sau funcţii apelate de către program, selectată dintr-o listă de rutine apelate printr-o linie cursor.

Find procedure. Crează un cadru de dialog, permiţînd precizarea unui nume de procedură sau funcţie care vą fi încărcată în editor.

Integrated debugging. Indică generarea de informaţie de depanare necesară folosirii depanatorului integrat.

Standalone debugging. Informaţiile de depanare generate de compilator vor fi înglobate în fişierul EXE creat pe dischetă.

Are sens în cazul folosirii unui depanator extern independent.

Display swapping. Precizează modul în care depanatorul afişează ecranul de execuţie.

Smart — ecranul de execuţie apare numai dacă instrucţiunea modifică afişarea.

Always — ecranul de execuţie apare după fiecare instrucţiune.

None — ecranul de execuţie este afişat numai la acţionare ⟨Alt⟩ ⟨FS⟩.

Refresh display — reface complet afişarea mediului de dezvoltare.

Meniul BREAK/WATCH

Cînd se lucrează cu depanator integrat, meniul Breat/Watch permite adăugarea sau ştergerea de variabile martor din fereastra Watch şi plasarea şi ştergerea de puncte de interpretare. (fig. 7)

```
Add watch                Ctrl-F7
Delete watch
Edit watch
Remove all watches
---------------------------------
Toggle breakpoint        Ctrl-F8
Clear all breakpoints
View next breakpoint
```

figura 8

Add watch — permite adăugarea unei noi variabile martor (Watch) în fereastra Watch; cînd aceasta este activă se pot adăuga variabile martor folosind testele de editare ⟨**Ctrl**⟩ ⟨**N**⟩ sau ⟨**Ins**⟩.

Delete watch — permite ştergerea unei variabile sau expresii martor din fereastra Watch; cînd aceasta este activă, acelaşi rezultat se obţine cu tastele ⟨**Ctrl**⟩ ⟨**Y**⟩ sau ⟨**Del**⟩.

Edit watch — permite editarea numelui unei variabile sau a unei expresii martor aflată în fereastra Watch.

Remove all watches — şterge toate variabilele şi expresiile martori din fereastra Watch.

Toggle breakpoint — plasează sau şterge un punct de întrerupere în linia în care se afla cursorul.

Clear all breakpoint şterge toate punctele de întrerupere.

View next breakpoints — mută cursorul pe următorul punct de întrerupere.

4. Facilităţi ale limbajului TURBO Pascal versiunile 5.0 şi 5.5.

Faţă de versiunile anterioare sistemul TURBO Pascal versiunea 5.0 prezintă următoarele avantaje:

1) depanarea la nivelul codului sursă, incluzînd: execuţia pas cu pas, puncte de întrerupere, examinarea variabilelor, structurilor de date şi expresiilor;

2) emularea virgulei mobile în lipsa coprocesorului matematic 8087 sau 80287;

3) segmentarea bazală pe unităţi;

4) compatibilitate cu TURBO Debugger;

5) posibilitatea folosirii extensiei de memorie (de către editorul de texte şi de către unitatea Overlay);

6) viteza de compilare de 3—5 ori mai ridicată faţă de versiunea 3.0;

7) generarea de cod îmbunătăţită producînd o execuţie mai rapidă;

8) editor de legături încorporat care înlătură porţiunile nefolosite de cod şi date producînd programe mai mici;

9) fişiere EXE producînd programe care depăşesc 64 KOcteţi de memorie;

10) compilare separată folosind unităţile de memorie;

11) unităţi standard: System, Dos, Crt, Overlay, Printer şi Graph;

12) interfaţa mai puternică cu limbajul de asamblare;

13) posibilitatea de includere a fişierelor pe 8 niveluri de adîncime;

14) tipuri de date noi: shortint, longint, word, single, double, extended, comp;

5) proceduri şi funcţii standard noi;

16) evaluarea optimizată a expresiilor booleene;

17) directive de compilare condiţionată;

18) compatibilitate ridicată cu versiunile 3.0 şi 4.0; utilitare pentru conversia programelor între versiuni;

19) două versiuni de compilator: mediu integrat de dezvoltare (editor, compilator, editor de legături, depanator, biblioteca runtime) şi compilator cu linie de comandă;

20) recompilarea tuturor unităţilor (opţiunea Build) sau în mod selectiv după dată (opţiunea Make) la efectuarea unei actualizări într-una din unităţi sau în program;

5. Diferenţe între versiunile 5.0—5.5 şi 3.0

VERSIUNEA 3.0	VERSIUNILE 5.0 - 5.5
Un identificator din program poate avea acelaşi nume cu numele programului	numele programului este unic; orice identificator din program poate fi referit prin nume program. identificator
toate obiectele programului se compilează în acelaşi timp extrăgîndu-se dintr-un fişier sursa şi din fişiere incluse	obiectele se grupează în unităţi care se compilează separat
un program executabil (fişier .COM) poate avea maxim 64 KO cod	programul executabil (.EXE) este format din unităţi, fiecare putînd avea pînă la 64 KO cod
pentru a depăşi restricţiile de memorie se foloseşte segmentarea şi înlănţuirea	facilităţile de segmentare sînt mai sofisticate şi mai transparente pentru utilizator. Inlănţuirea programelor se face cu procedura Exec din unitatea Dos
opţiunea de includere ($I nume poate fi plasată oriunde şi poate fi formată numai din instrucţiuni	fişierul inclus nu poate conţine numai instrucţiuni ci proceduri şi funcţii complete; el nu poate fi plasat între begin şi end

fișierele incluse nu pot include alte fișiere	sînt permise incluziuni de fișiere pînă la 9 niveluri de adîncime
intrările de la consola sînt tratate nestandard comparativ cu intrările disc	intrările consola sînt compatibile cu intrările din fișierele disc
	tipuri noi: shortint, longint, word, single, double, extended, comp
constantele predefinite sînt cunoscute în tot programul	constantele predefinite în unități sînt accesibile din program numai dacă se folosește clauza uses
IOResult întoarce coduri de eroare specifice versiunii 3.0	IOResult întoarce coduri de eroare specifice MSDOS.
tipuri string numai cu indicație de lungime	string = string (255)
într-o procedură sau funcție variabilă de control a ciclului for poate fi locală sau globală	într-o procedură sau funcție variabila de control a ciclului for trebuie să fie locală
se pot declara etichete fără a le folosi	etichetele declarate și nefolosite sînt erori sintactice
lungimea variabilei tampon pentru fișiere text poate fi stabilită la declararea variabilei fișier	lungimea variabilei tampon pentru fișiere text poate fi stabilită numai cu SetTextBuffer
constanțele cu tip sînt rezidente în segmentul de cod	constanțele cu tip sîn rezidente în segmentul de date
conversie numai între tipurile ordinale	conversie între toate tipurile simple cu restricția ca sursa și destinația să aibă aceeași lungime
evaluarea completă a expresiilor booleene	evaluarea optimizata a expresiilor booleene
proceduri de tratare a întreruperilor numai în limbaj de asamblare	proceduri de tratare a întreruperilor în Pascal folosind directiva interrupt
rutinele externe în limbaj de asamblare trebuie să fie în format .BIN cu ofseturi față de prima rutină din fișier	rutinele în limbaj de asamblare pot fi în format .OBJ fiind legate folosind opțiunea $L fișiere-obiect
apelurile de funcții si proceduri sînt apeluri scurte în acelaşi segment de cod	se generează după necesitate apeluri scurte sau lungi $F+ forțează apeluri lungi între segmente
tabelul 2	instrucțiunea inline nu permite referiri la contorul de locații nici la nume de proceduri sau de funcții

IMPORTANT!

Am primit la redacție: S-a constituit — *Fundația „East-West Education Development"*, sponzorizată de Patrick McGovern - Președintele IDG - și de către IDG. Scopul principal: sprijinirea țărilor din estul Europei privind accesul la tehnologia informației în îmbunătățirea sistemului de învățămînt și de educație prin intermediul calculatoarelor.

Fundația roagă toți producătorii și comercianții de tehnică de calcul să pună la dispoziția acesteia echipamentele la care renunță. Avantajele sînt, credem, cunoscute: scutirea de impozite, reclamă etc. Vă rugăm să ne răspundeți pe adresa redacției, noi urmînd a trimite fundației răspunsul dumneavoastră. Vă mulțumim!

Computerworld

Recent a fost difuzat un raport cu privire la securitatea informațiilor și a datelor vehiculate prin intermediul calculatoarelor, avînd în vedere proliferarea rețelelor. Raportul, intitulat sugestiv „Computers at risk" a concluzionat că utilizatorii și comercianții din domeniu nu sînt încă suficient de bine informați și nu acordă importanța cuvenită sistemului de protecție comercială. În același timp, raportul a pus bazele unui sistem de protecție a informațiilor (GSSP — Generally Accepted System Security Principles) care reprezintă un ghid de protecție pentru utilizarea calculatoarelor în rețele. Acest raport a fost conceput de un comitet pentru studierea sistemelor de securitate, afiliat la Academia Națională de Științe din S.U.A. ca un răspuns la cererea formulată de DARPA (Defense Advanced Research Projects Agency). Din acest comitet fac parte 16 membri implicați în producția sau cercetarea din domeniu de la instituții importante, dintre care: AT and T Bell Laboratories, BBN Communications Corp. Digital Equipment Corp. și altele.

Iată cîteva dintre recomandările raportului, în afara celor menționate mai sus:

— determinarea utilizatorilor și distribuitorilor de a utiliza cît mai repede și eficient sistemele deja existente de securitate;

— alcătuirea unei baze de date cu privire la toate informațiile existente despre securitatea datelor și tranzacțiilor și încurajarea specializării în aceste sisteme de securitate și de... etică profesională;

— simplificarea și sistematizarea restricțiilor guvernamentale existente cu privire la comercializarea sistemelor de securitate și codificare a datelor peste ocean.

— înființarea unui fond special pentru cercetările în domeniul securității informațiilor și protecțiilor. (Network World/ 3 decembrie 1990/ 7 ianuarie 1991, Digital News/ 7 ianuarie 1991).

Computerworld

NeXT DIN NOU ÎN ACTUALITATE

Traducere și prelucrare de Mihaela GORODCOV după articolul „NEXT ON THE AGENDA“ de Bruce F. Webster apărut în numărul din ianuarie 1991 al revistei MACWORLD.

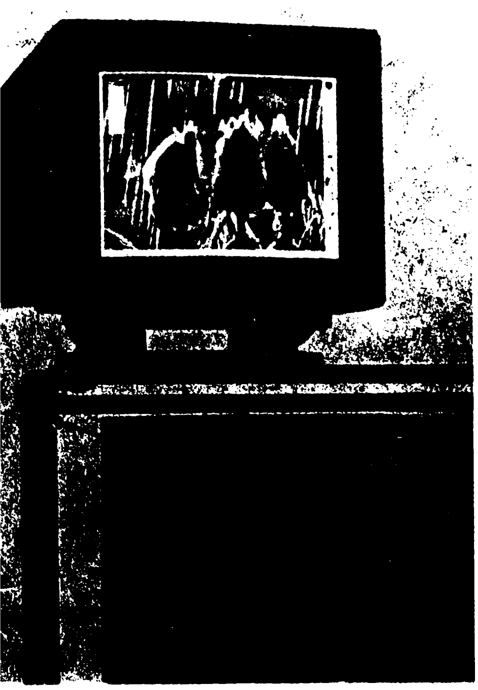
Cu doi ani în urmă, anunțam în paginile revistei „Știință și tehnică“ și ale almanahului cu același nume, apariția în „forță“ a unui nou tip de calculator cu multe inovații, care se anunța deosebit de promițător: sistemul NEXT. Steve Jobs — unul dintre „copiii teribili“ ai informaticii și cofondatorul — alături de Steve Wozniak, al firmei Apple, a creat NeXT-ul ca o mașină îndrăzneață, lansată cu o campanie publicitară corespunzătoare în octombrie 1988. Mănușa fusese aruncată! NeXT-ul, cel controversat, urma să revoluționeze din multe puncte de vedere modul de lucru „personal“ al utilizatorului care avea acum la dispoziție o mașină puternică, cu multe facilități și deschideri spre numeroase aplicații.

Firește că la NeXT se regăseau multe dintre posibilitățile calculatoarelor Apple Macintosh, precum și ale altora. Desigur că NeXT-ul avea, de asemenea, limitările sale: lipsa software-ului comercial, prețuri destul de ridicate și lucrul destul de lent. Asupra altor caracteristici ale NeXT-ului din octombrie 1988 nu mai revenim. Ele au fost descrise pe larg în revista și în almanahul ST.

Acum, după doi ani de la lansare, nimeni nu poate spune că NeXT nu a „învățat“ ceva din propriile greșeli. Revenirea din septembrie 1990 a fost făcută tot „în forță“. Compania a introdus o linie de sisteme care nu numai că sînt mai puternice, dar și mai... ieftine decît mașina inițială. Înalta rezoluție și larga paletă de tonuri de gri a monitoarelor alb/negru ale modelului inițial a fost acum completată cu două sisteme color. Mai mult decît atît, două dintre companiile faimoase de software și-au introdus produsele lor pe NeXT: IMPROV (LOTUS) și WORDPERFECT de la firma cu același nume. Si, pentru a liniști „vocile“ care acuză numărul relativ mic de mașini instalate, NeXT a anunțat că are deja comenzi ferme pentru 15 000 de

MAC IIFX VERSUS NEXTCUBE

	Mac IIfx System	List Price	Nextcube System	List Price
Basic System	4MB RAM, 160MB hard drive	$10,969	8MB RAM, 340MB hard drive	$9000
CPU	40MHz 68030		25MHz 68040	
MMU	built-in to 68030		built-in to 68040	
FPU	40MHz 68882		built-in to 68040	
DSP	none		25MHz 56001	
I/O processors	1 SCSI DMA, 2 serial I/O processors		12 DMA I/O processors	
ROM	512K		NA	
RAM	16MB	$880	16MB	$400
Video	Macintosh Display Card 8•24	$899	1120 × 832 × 4 shades (built-in)	
Monitor	Apple Two-Page Monochrome Monitor (1152 × 870 × 16)	$2149	MegaPixel Display (1120 × 832 × 4)	$995
Keyboard	Apple Keyboard	$129	Next keyboard	
Sound input	none		8-bit, 8kHz sampling	
Sound-input ports	none		microphone jack, built-in microphone	
Sound output	Apple sound chip with built-in speaker		56001 DSP with built-in speaker	
Sound-output ports	stereo minijack		stereo minijack, dual line-outs	
Other ports	serial (2); SCSI; ADB (2)		serial (2), SCSI/2, printer, DSP, Ethernet	
Slots	6 (4 available)		4 (3 available)	
Floppy drive	1.44MB		2.88MB	
Networking	AppleTalk (built-in), EtherTalk	$699	Ethernet (built-in)	
System software	A/UX 2.0 (on floppies)	$995	Release 2.0 extended	
Bundled software	System 6.0.6, HyperCard, Edit, Shell, UNIX utilities		WriteNow, Webster's Ninth New Collegiate Dictionary, Librarian, Mail, Edit, Shell, UNIX utilities, Quotations, Shakespeare, demo applications, NextStep development tools	
Total Retail Price for Monochrome System		$16,720		$10,395
Color monitor	Radius 19" Display (1152 × 882)	$4295	Next MegaPixel Color (1120 × 832)	$2995
Color video board	Radius DirectColor/24	$3595	Nextdimension (32-bit color with alpha)	$3995
Graphics acceleration	Radius QuickColor	$595	Intel i860 RISC processor	
Video integration	RadiusTV	$2795	NTSC/S-Video/RGB input and output	
Color compression	color-compression board [Note: no slots available]	$995	C-Cube JPEG color compression chip [note: two slots available]	
Total Retail Price for 24-bit System		$28,995		$16,390



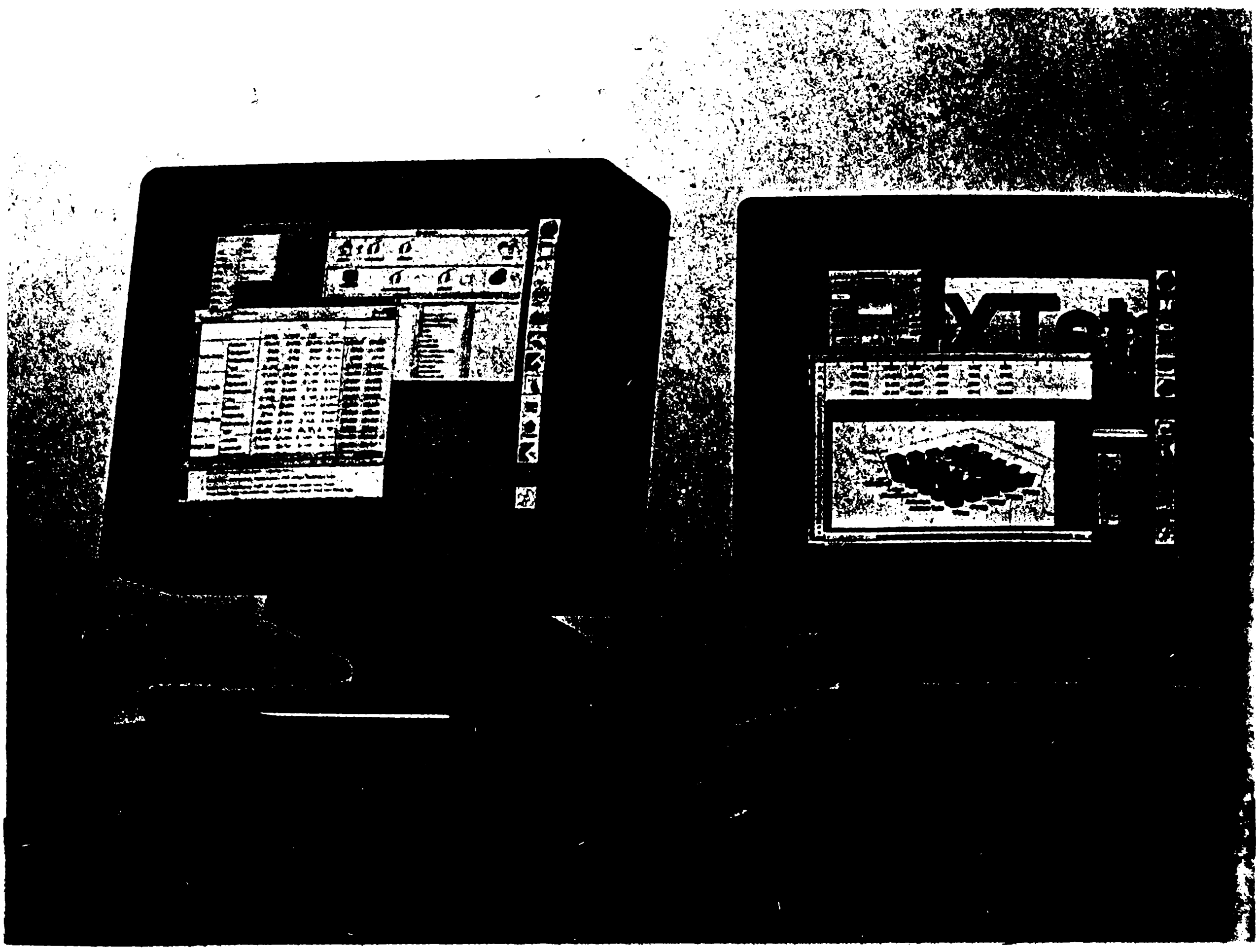
sisteme.

Dar să vedem, pe scurt, care sînt membri familiei NeXt. Au fost lansate deja trei sisteme de bază: NEXTSTATION, NEXTSTATION COLOR și NEXTCUBE la care se pot conecta cîeva echipamente deosebit de performante: MegaPixel Display, imprimanta laser NeXT cu 400 dpi (dot per inch) — la un preț redus —, extinsă pentru grafică video Nextdimension pe 32 biți și monitorul color de 16 inch MegaPixel Color Display.

Toate cele trei sisteme NeXt întrunesc calități și performanțe cu adevărat deosebite:

- Microprocesor 68040 (Motorola) de 25 MHz pe post de CPU (Central Processing Unit), unitatea de gestionare a memoriei (MMU — Memory Management Unit) și unitatea de virgulă mobilă (FPU — Floating Point Unit).
- Procesor pentru prelucrarea digitală a semnalului (DSP) 56001 la 25 MHz cu 24 K memorie în RAM Static expandabilă pînă la 576 K.
- 16 conectoare SIMM pentru memoria principală capabile să accepte de la 1MB pînă la 4MB.
- 2 cipuri VLSI specializate pentru „mainframe" cu rolul de a implementa 8 procesoare I/0 (9 pentru Next Cube) și alte facilități de sistem.
- Placă video și RAM — video separat — care generează imagini de 1120 pixeli pe 832 linii, care constituie suportul pentru „alpha-channel".
- Includerea hardware-ului Ethernet specializat cu conectoarele externe atît pentru BNC T-connector, cît și pentru cablul panglică 10 Base-T, acesta din urmă fiind capabil să suporte conexiuni Ethernet de la fire standard telefonice.
- Un drive de flopyy disc de 3 1/2 inch cu capacitate de 2,88 MB care poate, de asemenea, citi și înscrie discuri MS-DOS (atît de 720 K cît și de 1,44 MB)
- Porturi duale seriale (DIN-8 RS-422); porturi specializate pentru video și imprimantă; port DSP; interfață-SCSI/2 cu porturi interne și externe.
- Intrare de sunet (inclusă în interiorul mașinii) de 8 biți și 8 KHz, eșantionare și ieșire de sunet — cu canal dublu de 16 biți și 44,1 KHz eșantionare.
- „Release 2.0." ca sistem software al lui NeXT care include Next Step 2.0. aflat în topul sistemului de operare Mach (UNIX).

Cam acestea ar fi în mare facilitățile și performanțele familiei NeXT, care deschid — după cum se poate observa — un cîmp foarte larg de aplicații multor categorii de utilizatori. În cele ce urmează vă propunem o foarte succintă „fișă tehnică" a fiecărui sistem, cu alte cuvinte cîteva tabele comparative între mașinile Next și cele din familia Macintosh Apple. Desigur că nu ne propunem în acest material să epuizăm subiectul NEXT. El este foarte vast și, firește, că vom reveni asupra lui. De asemenea, cu ocazia acestui material facem și o deschidere către familia de calculatoare Apple Macintosh, căreia, în numerele viitoare îi vom consacra niște materiale speciale din documentația „la zi" primită prin rețeaua IDG.



Computerworld

	Mac IIsi System	List Price	Nextstation System	List Price
Basic System	2MB RAM, 40MB hard drive	$3800	8MB RAM, 105MB hard drive	$4000
CPU	20MHz 68030		25MHz 68040	
FPU	20MHz 68882	$200	built-in to 68040	
MMU	built-in to 68030		built-in to 68040	
DSP	none		25MHz 56001	
I/O processors	none		12 DMA I/O processors	
ROM	512K		NA	
System RAM	8MB	$600	8MB	
Built-in video	640 × 480 × 256 colors/shades 640 × 870 × 16 shades		1120 × 832 × 4 shades (plus 4 levels of alpha)	
Monitor	Apple Portrait Display (640 × 870)	$1099	MegaPixel Display (1120 × 832)	$995
Keyboard	Apple ADB Keyboard	$99	Next keyboard	
Sound input	8-bit, 11kHz or 22kHz sampling		8-bit, 8kHz sampling	
Sound-input ports	microphone jack (microphone included)		microphone jack, built-in microphone	
Sound output	Apple sound chip with built-in speaker		56001 DSP with built-in speaker	
Sound-output ports	stereo minijack		stereo minijack, dual line-outs	
Other ports	serial (2), SCSI, ADB		serial (2); SCSI/2, printer, DSP, Ethernet (thin wire and twisted pair)	
Slots	1		0	
Floppy drive	1.44MB		2.88 MB	
Built-in networking	AppleTalk		Ethernet	
System software	System 6.0.6		Release 2.0	
Bundled software	HyperCard, System utilities		WriteNow, Webster's Ninth New Collegiate Dictionary (small version), Librarian, Mail, Edit, Shell, UNIX utilities	
Total Retail List Price		$5798*		$4995

** With AppleColor High-Resolution RGB Monitor instead (640× 480), $5698.*

	Mac IIci System	List Price	Nextstation Color System	List Price
Basic System	4MB RAM, 80MB hard drive	$6669	12MB RAM, 105MB hard drive	$4875
CPU	25MHz 68030		25MHz 68040	
MMU	built-in to 68030		built-in to 68040	
FPU	25MHz 68882		built-in to 68040	
DSP	none		25MHz 56001	
I/O processors	none		12 DMA I/O processors	
ROM	512K		NA	
RAM	8MB	$300	12MB	
Built-in video	640 × 480 × 256 colors		1120 × 832 × 4096 colors (with 4-bit alpha)	
Monitor	AppleColor High Resolution RGB Monitor (640 × 480)	$999	Next MegaPixel Color (1120 × 32)	$2995
Keyboard	Apple Keyboard	$129	Next keyboard	
Sound input	none		8-bit, 8kHz sampling	
Sound-input ports	none		microphone jack, built-in microphone	
Sound output	Apple sound chip with built-in speaker		56001 DSP with Next Sound Box	$125
Sound-output ports	stereo minijack		stereo minijack, dual line-outs	
Other ports	serial (2), SCSI, ADB (2)		serial (2), SCSI/2, printer, DSP, Ethernet (thin wire and twisted pair)	
Slots	3		0	
Floppy drive	1.44MB		2.88MB	
Built-in networking	AppleTalk		Ethernet	
System software	System 6.0.6		Release 2.0	
Bundled software	HyperCard, System utilities		WriteNow, Webster's Ninth New Collegiate Dictionary (small version), Librarian, Mail, Edit, Shell, UNIX utilities	
Total Retail List Price		$8097		$7995

Virgil IONESCU
Radu CONSTANTINESCU

BRIGHT PENTRU HC-85

Deşi HC-85 este calculatorul românesc care reproduce cel mai fidel posibilităţile lui Sinclair Spectrum, utilizatorii săi şi-au putut da seama încă din primele momente că el s-a născut „bolnav", din punct de vedere hard, BRIGHT nefiind implementat. Această infirmitate ne privează, pe de o parte, la crearea unor programe de utilizarea şi a nuanţelor derivate din cele opt culori, iar pe de altă parte, la rularea programelor de firmă, de gama integrală a informaţiei. Un exemplu edificator în acest ultim caz, este imposibilitatea folosirii grilei de BRIGHT în utilitarul „Artstudio".

Vă propunem remedierea acestei deficienţe prin utilizarea la maximum a circuitelor plăcii de bază. Trebuie remarcat că, suplimentar, se folosesc: un tranzistor NPN, trei rezistoare şi circa 0,5 m sîrmă...

Cele de faţă constituie şi o invitaţie adresată producătorului, ICE, pentru revizuirea plăcii de bază în sensul includerii schemei BRIGHT prezentate. Figura 1 reprezintă schema BRIGHT. Conexiunea între părţile anterior nefolosite ale circuitelor integrate se realizează cu sîrmă izolată pe partea cu piese, exceptînd conexiunea 11 (El) — 3 (E2) care poate fi făcută pe partea opusă. Lipiturile se fac direct la pinii circuitelor integrate. Este bine să se respecte următoarea ordine:

Pin C.I. Pin C.I.
1 (E2) — 15 (A1)
1 (E2) — 12 (C8)
6 (D7) — 13 (C8)
14 (C8) — GND (de ex. 15 (C7))

Tranzistorul şi cele trei rezistoare vor fi implantate în zona A6 a plăcii, conform schemei din figura 2. Este preferabil să se folosească rezistoare cu peliculă metalică sau chimică de 0,25 W (datorită dimensiunilor reduse). Tranzistorul poate fi BC 107, 108, 109 etc.

A1, C7, C8, D7, E1, E2, R110, R111, R112, R113, T9 sînt denumirile componentelor din schema producătorului.

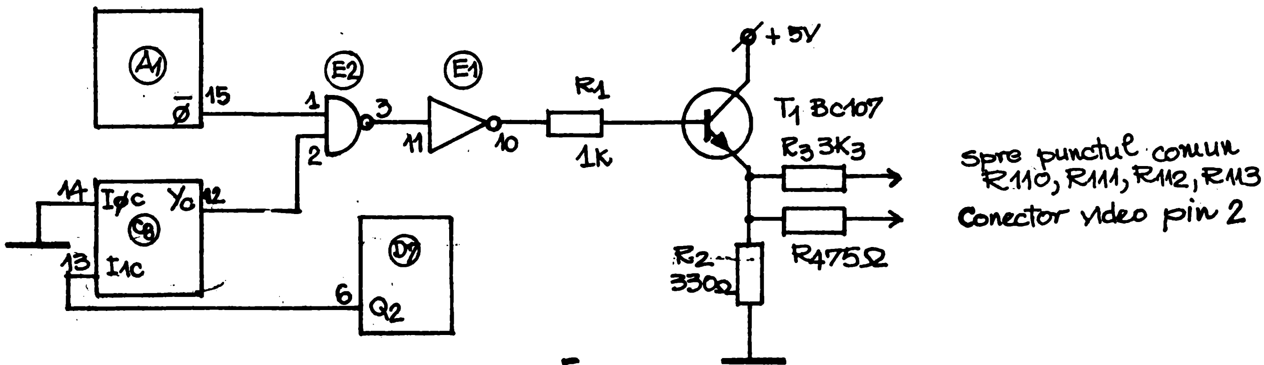


Fig. 1

Funcţionarea schemei. Semnalul de BRIGHT, bitul 6 al octetului de atribute, este prezent pe pinul 6 al registrului de atribute D7. Prin C8 semnalul este multiplexat cu cel de BORDER şi este trimis la poarta E2, tip NAND. Semnalul de BRIGHT este oprit dacă culoarea curentă este 0 (nu există BRIGHT pe negru la Spectrum!) cu ajutorul decodificatorului de culoare din codorul PAL (A1). În continuare, semnalul este inversat prin E1, amplificat de tranzistorul T si aplicat în punctul comun al rezistoarelor R110-R113, unde se formează semnalul Y de luminanţă.

Pentru monitoare RGB semnalul trebuie preluat din emitorul tranzistorului T printr-o rezistenţă de 75 ohmi şi aplicat la conectorul video (pinul 2).

Trebuie precizat că la conectarea schemei BRIGHT nivelul de luminanţă va scădea puţin, ceea ce poate fi eventual compensat printr-o uşoară modificare a reglajului de nivel din modulatorul TV.

Se recomandă ca modificările prezentate să fie executate de un specialist în domeniu, pentru a nu avea surprize cel puţin neplăcute!

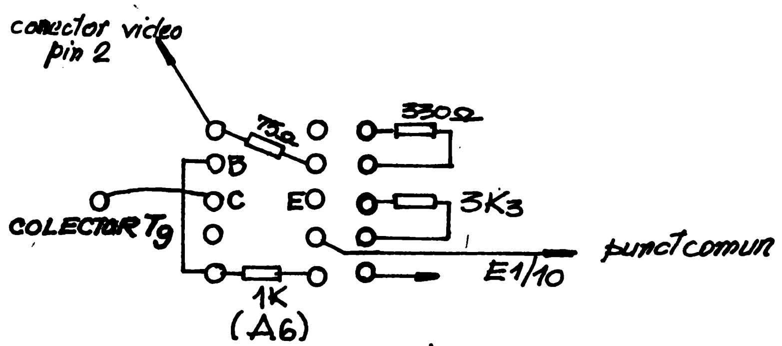


Fig. 2

Ozias ADLER

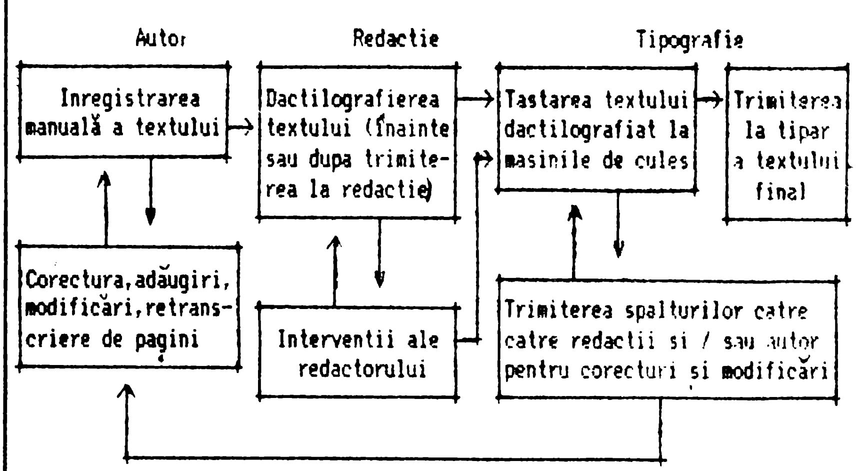
UN INSTRUMENT DE LUCRU ÎN REDACȚIA NOASTRĂ : CALCULATORUL!

Desigur că, multora dintre dv. acest articol și, mai ales, microcalculatoarele CP/M, li se vor părea depășite și totuși.... Noi îl publicăm, după mai bine de 3 ani, ca un semn de restituire și respect pentru munca unor oameni entuziaști, care au reușit să facă din redacția „Știință și tehnică", cu toate avatarurile acelor vremuri, prima redacție informatizată din România. Pentru această prioritate și pentru un imens capital sentimental, publicăm acum acest articol, cu toate mulțumirile și recunoștința noastră pentru cei care ne-au ajutat și au crezut în noi.

Mulți utilizatori de microcalculatoare sînt convinși că un editor de texte gen Wordstar satisface orice cerință. Și acestora le sînt adresate explicațiile ce urmează pentru a putea face diferența dintre textele culese în tipografie și cele realizate după programele de tip Wordstar.

Orice text - indiferent de conținutul său - își începe viața de-abia după ce este înregistrat. Operația se poate face pe hîrtie sau pe suport magnetic, dar rămînerea la acest stadiu face din text numai un document primar, chiar și în cazul cînd, în continuare, el este multiplicat prin cele mai diferite mijloace. Majoritatea textelor ajunse la cititor reprezintă însă rezultatul unor îndelungate și repetate prelucrări ce se constituie într-un domeniu cunoscut, dar în intensă și rapidă dezvoltare - activitatea redacțional-editorială.

Pînă în urmă cu cîteva decenii, instrumentele de lucru din redacții erau dintre cele mai simple: creionul (stiloul, pixul), guma, foarfecele și lipiciul. Un instrument ceva mai evoluat era mașina mecanică de scris, în fața căreia, de multe ori, nu se află un redactor, ci doar o dactilografă. Productivitatea muncii cu asemenea mijloace rudimentare este destul de redusă, mai ales dacă se ține seama că un text pornit de la sursă (autorul) suferă cîteva reînregistrări pînă cînd ajunge la destinație (cititorul). Vechiul proces redacțional este reprezentat în schema nr. 1.



Schema 1 - Vechiul proces redactional

După cum se observă, operațiile cele mai redundante sînt înregistrarea textului și corecturile cu un coeficient de repetare de pînă la 2,5-3 ori. În plus, procesul de transmitere a textului de la autor la cititor este mult încetinit în cazul editărilor de cărți și reviste de multiplul schimb de corecturi dintre tipografii și redacții.

Ce nu știe cititorul

Un text destinat tiparului nu poate fi transmis ca atare tipografiei. El trebuie însoțit de o serie de indicații de tehnoredactare din care enumerăm pe cele mai importante:

● **formatul de bază al rîndurilor** (exprimat în unități de măsură vechi, specifice poligrafiei, de exemplu cicero sau pica, puncte tipografice sau unități relative, cvadrați etc.);

● **familia de litere** (aceasta este de obicei un anume sortiment de semne de tipar elaborat de un grafician specializat în proiectare de alfabete pentru tipar; ele pot fi protejate prin legea dreptului de autor și poartă nume distincte, de exemplu univers, megaron, bodoni, gill, souvenir etc.);

● **tăietura literei** (este o specificare suplimentară a unei variante a familiei de litere, de exemplu univers drepte, cursive, aldine, subțiri, aldin-cursive etc.);

● **corpul de literă** (este înălțimea literei, exprimată în puncte tipografice);

● **interlinierea** (este măsura distanței dintre rînduri, exprimată în puncte și fracțiuni de puncte tipografice);

● **starea** (mărimea spațiului fix cu care începe un paragraf sau aliniat exprimat în pătrișori, semipătrișori, puncte tipografice);

● **specificațiile** pentru titluri, legende la figuri, note de picior etc.

Toate aceste indicații modelează în fel și chip aspectul textului tipărit și reprezintă componente esențiale ale prezentării poligrafice ce poate fi numită și estetica de carte.

Un text tipărit nu poate fi pus pe picior de egalitate cu un text dactilografiat sau listat de imprimantă din considerente de estetică a tiparului (chiar și în cazul celor mai avansate sisteme de birotică, de exemplu Desktop Publishing). Varietatea familiilor și corpurilor de literă este deosebit de bogată în cazul tiparului și extrem de săracă la textele dactilografiate sau listate. Lizibilitatea unui text tipărit este net superioară, mai ales din cauza variației lățimilor literelor (la tipar aproape fiecare literă poate diferi de alta prin lățime, în timp ce litera sau semnul dactilo sau listat au lățimi egale - de exemplu literele m și i, precum și punctul sau virgula au lățimi egale între ele). În plus, spațiile dintre cuvinte (blancurile) variază la tipar imperceptibil, aproape fără trepte de la rînd la rînd, dar au mărimi constante în cadrul aceluiași rînd. Dar... să ne oprim din argumentare, căci criteriile de estetică a tiparului nu pot fi epuizate în cîteva exemple și, de altfel, nu acesta este obiectul articolului de față.

Calculatoarele electronice au adus și în cadrul redacțiilor modificări substanțiale ale stilului de lucru și aceasta, trebuie să recunoaștem, constituie, prin efortul de a se adapta, o dificultate pasageră pentru redactorii zilelor noastre, dar sporește cu mult productivitatea celor implicați în activitatea redacțional-editorială.

Implementarea lor în redacții nu ar fi fost posibilă dacă în domeniul culegerii textelor pentru tipar nu ar fi apărut fotoculegerea - procedeu tehnologic ce înlocuiește culegerea cu plumb pe linotip, monotip etc. cu culegerea pe hîrtii sau filme fototehnice prin intermediul unor automate de expu-

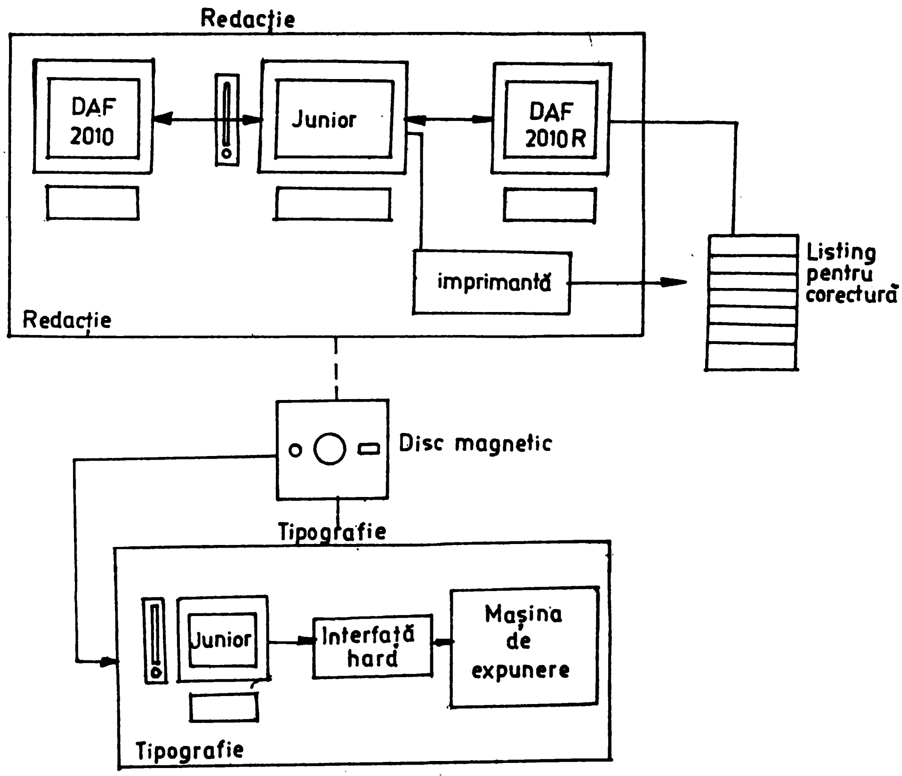


FIGURA 1

nere a textului. Practic, toate acestea au integrate calculatoare de proces specializate, iar productivitățile lor variază de la zeci de mii pînă la multe milioane de semne pe oră în funcție de generația căreia îi aparțin (și în fotoculegere specialiștii se referă la un număr de 4–5 generații, la fel ca în domeniul calculatoarelor de uz general!).

Calculatorul intervine

Comparativ cu prima schemă, activitatea redacțională asistată de calculator se prezintă ca în schema nr. 2. Este ușor de remarcat că nu mai sînt necesare înregistrările repetate ale textului condiționate de modificări multiple (intervenții, adăugiri și eliminări masive) și alte tipuri de corecturi, sau de prelucrarea textului pe mașinile de cules. De asemenea se observă că întregul proces de corectură (semnalarea și eliminarea erorilor) se face în interiorul redacției și că tipografia nu mai poate influența, datorită schimbului de corecturi, termenele și prioritățile de apariție.

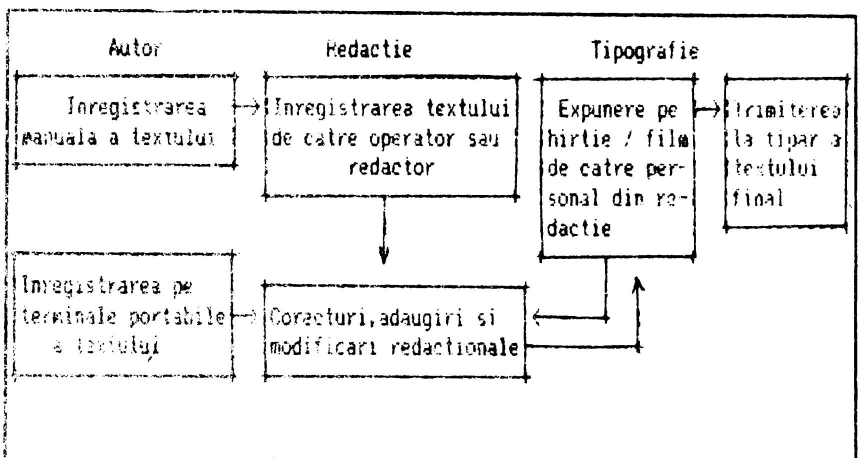
Sistemul redacțional-poligrafic ce funcționează în cadrul redacției revistei „Știință și tehnică" este bazat pe microcalculatoare pe 8 biți compatibile CP/M și este format exclusiv din componente de tehnică de calcul produse de IEPER-București.

În figura 1 sînt prezentate configurația acestui sistem, fluxul tehnologic al activității redacționale și legătura cu mașinile de expunere ale tipografiei Combinatului Poligrafic din Capitală. Componentele configurației au fost alese, din considerente tehnologice și de eficiență economică, din sortimentul de produse al întreprinderii sus amintite oferit la data elaborării programului. Locul de muncă al operatorului, corectorului sau redactorului la înregistrarea textului este videoterminalul DAF 2010R – varianta poligrafică.

Microcalculatoarele Junior sînt folosite ca servere de fișiere pentru terminalele de culgere DAF 2010R. Fișierele pot fi atît programele de funcționare ale DAF-urilor, cît și articole sau fragmente de text. Imprimantele, cuplate și ele la aparatele Junior, produc listing-uri cu un set extins de caractere (circa 128), cu care se poate face culegerea de texte în limbile română, franceză, engleză, germană, italiană, spaniolă, maghiară; este identic cu setul implementat pe DAF-uri.

Un alt echipament Junior, plasat în tipografie, preia discurile cu textele finale obținute în redacție și le transmite mașinilor de expunere prin intermediul interfeței hard și al programului de transmisie SEND, elaborate de un specialist al IEPER – ing. Gabriel Dulcu.

Pachetul de programe de calcul pentru activitatea redacțional-poligrafică a fost elaborat de către autorul articolului de față. Două dintre acestea vor fi prezentate în continuare.



Schema 2 – Noul proces redactional

Program dispecer în redacție

RIPALL este implementat pe Junior și asigură conducerea sistemului aflat în redacție. Un meniu permite alegerea unuia din cele două canale de legătură cu videoterminalele DAF 2010R și conținutul transmisiei: a) transmiterea programului redacțional-poligrafic; b) recepția de texte înregistrate sau corectate și depunerea lor pe discuri flexibile și c) transmiterea textelor de pe discuri către DAF-uri, după o conversie prealabilă din codul TTS (folosit de mașinile de fotoculegere Harris 3300) în codul ASCII. Ultima operație poate fi însoțită și de transmiterea textului către imprimantă, sau această transmisie poate fi făcută numai către imprimantă.

Pe discuri, textele sînt înregistrate sub formă de fișiere în cod TTS și ele reprezintă articole întregi sau fragmente de articol.

Program pentru activitatea redacțional-poligrafică

PLST este un program destinat realizării activității redacțional-poligrafice. În prezent, el este compus din următoarele module: înregistrare și editare de text, prelucrare poligrafică, culegere tabelară și culegere contorizată.

Modul de înregistrare și editare are cîteva caracteristici mai deosebite ce merită menționate. Ecranul este împărțit în două zone: rîndul de stare afișat în negativ și zona de text afișată normal.

În timpul înregistrării, în rîndul de stare sînt actualizate, pe măsura scrierii în zona de text, valorile unor parametri poligrafici importanți ce trebuie să se afle în permanență în atenția operatorului, cum sînt formatul rîndului, familia, tăietura și corpul de literă, interlinierea, tipul de retrageri în text, iar în cazul culegerii tabelare și numărul coloanei de tabel în care se lucrează.

În rîndul de stare se semnalează, de asemenea, dacă se lucrează în regim de inserare de caractere sau nu; la o comandă se afișează fie numărul de caractere, fie numărul de rînduri și înălțimea (în puncte tipografice) ale textului final expus (în vederea machetării textului în pagină). Se afișează, de asemenea, în rîndul de stare numărul erorii comise la înregistrarea textului, precum și tipul de culegere specială în care se lucrează (culegere tabelară sau contorizată).

Lucrul din zona de text are și el cîteva trăsături specifice. Să menționăm în primul rînd culegerea cu respectarea regulii cuvîntului întreg (wrap word), cunoscută și la alte programe editoare de texte. Textul nu este compus numai din semnele ce vor apărea în final pe hîrtie sau film, ci este însoțit de comenzi poligrafice plasate acolo unde trebuie să înceapă efectul lor. La înregistrarea textului, cînd se scrie o comandă poligrafică din cele afișate în rîndul de stare, valoarea nouă a acesteia este actualizată și în poziția comenzii în rîndul de stare. La înregistrarea textului singura comandă de corectură este ștergerea ultimelor caractere scrise. În general, adevărata corectură se face prin intrarea în regim de editare, care se obține automat prin deplasarea cursorului în interiorul textului. Sînt posibile operațiile elementare de corectură, caracter cu caracter, precum suprapunere de caracter, inserare sau eliminare de caracter, în ambele cazuri cu respectarea regulii de wrap word.

Operațiile de corectură mai complexă se efectuează pe blocuri de lungime variabilă. Una dintre acestea este anularea de blocuri de text. Prima comandă definește începutul blocului, iar cea de-a doua sfîrșitul, concomitent cu schimbarea afișării întregului bloc definit (de exemplu în video negativ). Întrucît există pericolul pierderii unor cantități mai mari de text prin manevrarea neatentă, comanda de anulare nu poate fi executată decît după confirmarea ei prin repetare. La prima comandă de anuare afișarea blocului se face în mod clipitor, tip de afișare ce se constituie astfel într-o formă de avertizare sau alarmare ce nu poate fi trecută cu vederea.

O altă corectură complexă este inserarea de blocuri de cîte 256 de semne fără modificarea afișării la fiecare caracter. Se creează o zonă de trei rînduri de blancuri, în care se înscrie cu mare rapiditate textul inserat. Repetarea comenzii înainte de epuizarea numărului de semne duce la îmbinarea sfîrșitului de bloc cu textul rămas, iar depășirea acestui număr conduce la inserarea de tip elementar, respectiv caracter cu caracter.

O a treia corectură complexă este mutarea de blocuri, cu sau fără anularea acestora în vechea poziție, și inserarea blocului în noua poziție. Despre corectura prin reîmplinire vom vorbi ceva mai încolo.

Textele pot fi afișate pe ecran în două moduri: neîmplinit sau împlinit. Textele împlinite sînt textele care formează o coloană în cadrul căreia lungimea rîndurilor de cuvinte este egală cu o anumită dimensiune (dată de tehnoredactor). Textele neîmplinite ocupă întreg ecranul. Modul neîmplinit este folosit de obicei la înregistrarea primară, cînd masivul de text nu a fost prelucrat conform comenzilor poligrafice care determină formarea unor rînduri ce corespund celor ce vor fi expuse pe materialele fototehnice.

După prelucrarea poligrafică textul poate fi afișat sub forma unor rînduri identice în conținut (nu și în lungime) cu rîndurile ce vor fi expuse pe mașina de fotoculegere. Eliminarea cîtorva caractere din aceste rînduri este permisă căci nu influențează aspectul rîndului expus. Inserarea de caractere cere de obicei corectura unor paragrafe (aliniate) prin reîmplinire. Pentru aceasta se efectuează corectura complexă cerută, se definește un bloc de text de mărimea unui paragraf sau unul care pornește de la începutul rîndului unde apare prima intervenție de corectură și se sfîrșește la capătul paragrafului. Corectura prin reîmplinire este foarte eficientă căci prelucrează numai fragmente reduse și nu întregi fișiere de text.

Tot în modul de înregistrare și editare se încadrează și tipurile de defilare a textului pe ecran (scrolling). Textul poate fi făcut să defileze rînd cu rînd sau pagină cu pagină, înainte și înapoi, prin repetarea comenzii sau poate fi afișat de la începutul sau sfîrșitul lui prin comenzi unice.

Modulul de prelucrare poligrafică are ca scop obținerea de rînduri împlinite (care prezintă pe ecran sau pe listing cu exactitate conținutul viitoarelor rînduri expuse). Pe ecran sau pe listing textul este însoțit și de comenzile poligrafice, în timp ce textul expus redă efectul lor.

În ce constă prelucrarea poligrafică?

După cum s-a mai spus, fiecare literă sau semn de tipar are de obicei lățimi care diferă între ele în cadrul aceleiași tăieturi, lățimi care au alte valori dacă se schimbă tăietura, corpul sau familia de litere. Ele sînt exprimate în unități relative.

Prima comandă poligrafică luată în considerare este formatul rîndului, care, prin calcul, este transformat din cicero în puncte tipografice și apoi în unități relative. Fiecare semn sau literă al cărei cod a fost prelucrat capătă o lățime recalculată și în funcție de tăietura, corpul și familia de litere cu care se lucrează. Lățimile sînt însumate în cuvinte, cuvintele sînt însumate în valoarea curentă a rîndului, ce este comparată cu valoarea dată a formatului. Distanțele dintre cuvinte (spații, blancuri), luate cu valoarea lor minimă, sînt și ele însumate la valoarea curentă a rîndului. După fiecare însumare, în rîndul curent se adaugă produsul dintre numărul de blancuri și valoarea extensiei admise pentru blancuri (stabilită în funcție de criterii de estetică a culegerii). Dacă prin această ultimă însumare se depășește formatul rîndului, atunci rîndul se poate împlini numai prin extinderea blancurilor, dacă nu se adaugă încă un cuvînt, iar dacă prin însumarea acestuia se depășește formatul, atunci ultimul cuvînt este introdus în rutina de despărțire în silabe pentru limba română. Aici, din coada cuvintelor sînt eliminate silabe și înlocuite cu cratime. Operația se termină atunci cînd partea de cuvînt rămas încape în formatul rîndului. Se plasează la capătul rîndului caracterul de sfîrșit de rînd, silabele eliminate sînt înscrise în noul rînd curent, care se împlinește ca și precedentul, și tot așa pînă la sfîrșitul textului. După terminarea fiecărui rînd se face conversia codurilor de text și de comenzi poligrafice, care sînt de tip ASCII, în coduri de tip TTS pentru mașina de fotoculegere.

Textul în coduri TTS este transferat către Junior și depus ca fișier pe discuri magnetice flexibile prin intermediul programului RIPALL.

Readucerea textului în memoria DAF-ului în vederea modificărilor de efectuat de către corectori, redactori sau conducătorii redacțiilor se poate face sub două forme, condiționate de mulțimea corecturilor de efectuat. Cînd numărul corecturilor este mare, cînd acestea sînt voluminoase sau cînd se schimbă parametrii culegerii, se optează pentru forma de prezentare neîmplinită. Cînd este vorba de corecturi puține ale unor caractere răzlețe sau corecturi mai mari, localizate în unele paragrafe, se alege forma împlinită. Corecturile sînt urmate de înregistrarea unor noi fișiere de text. Ciclul de corectură continuă pînă la obținerea produsului final corectat care este pe film. Semnalările de corectură făcute pe listing-ul de imprimantă de către corectori, redactori sau conducătorii redacțiilor sînt efectuate pe fișierele de text primare afișate pe DAF. Listing-urile cu text împlinit, corectat sau final pot servi și ca material de arhivare în vederea urmăririi fazelor editoriale.

Modul de culegere tabelară permite culegerea unor tabele cu pînă la 50 coloane într-un rînd. La trecerea de la o coloană la alta se modifică în rîndul de stare parametrii coloanei. Pe măsura culegerii cîte unui semn într-o coloană, valoarea formatului coloanei se reduce cu lățimea semnului cules pînă la epuizarea formatului, cînd culegerea în continuare este blocată în mod automat. Afișarea în rîndul de stare a variației restului de format permite operatorului să ia deciziile corespunzătoare la culegerea și la trecerea de la o coloană la alta. La terminarea rîndului tabelar, format din mai multe coloane, parametrii poligrafici ai primei coloane sînt din nou afișați în rîndul de stare.

Modulul de culegere contorizată vizualizează și el în rîndul de stare variația formatului unui rînd și blochează culegerea rîndului la epuizarea formatului. Cînd mărimea acestuia se reduce sub un sfert din valoarea lui inițială, literele din rînd încep să fie afișate în modul stabilit, iar cînd se intră în zona de împlinire cu ajutorul extensiilor de blancuri la acest tip de avertizare se adaugă și cel sonor. Ieșirea din starea de blocare se face prin ștergerea consecutivă a ultimelor semne pînă la locul unde se poate plasa o cratimă. După introducerea acesteia și a caracterului de sfîrșit de rînd, semnele șterse din rîndul precedent sînt readuse în noul rînd, bineînțeles cu readucerea și afișarea formatului curent în rîndul de stare. De asemenea, în rîndul de stare este afișată în permanență informația că operatorul lucrează în acest mod de culegere (lucru valabil și la culegerea tabelară).

Modul de culegere contorizată este folosit pentru culegerea unor rînduri răzlețe, pentru corecturi finale, dar este util și la culegerea unor texte în limbi străine pentru care nu s-au elaborat programe de despărțire în silabe, deciziile de terminare a rîndului fiind luate de operator.

În încheiere trebuie arătată că sistemul poate fi aplicat, fără nici un fel de modificări, și în actuala secție de fotoculegere din Combinatul Poligrafic din Capitală.

RUTINĂ GRAFICĂ pentru umplerea unor contururi

(urmare din pag.31)

Utilizare: rutina de umplere se apelează din BASIC de la adresa TEST, punctul de start fiind ultimul punct desenat cu PLOT sau PLOT INVERSE. Dacă se solicită umplere, punctul de start este marcat cu PLOT INVERSE 1, iar dacă se solicită ștergere, cu PLOT.

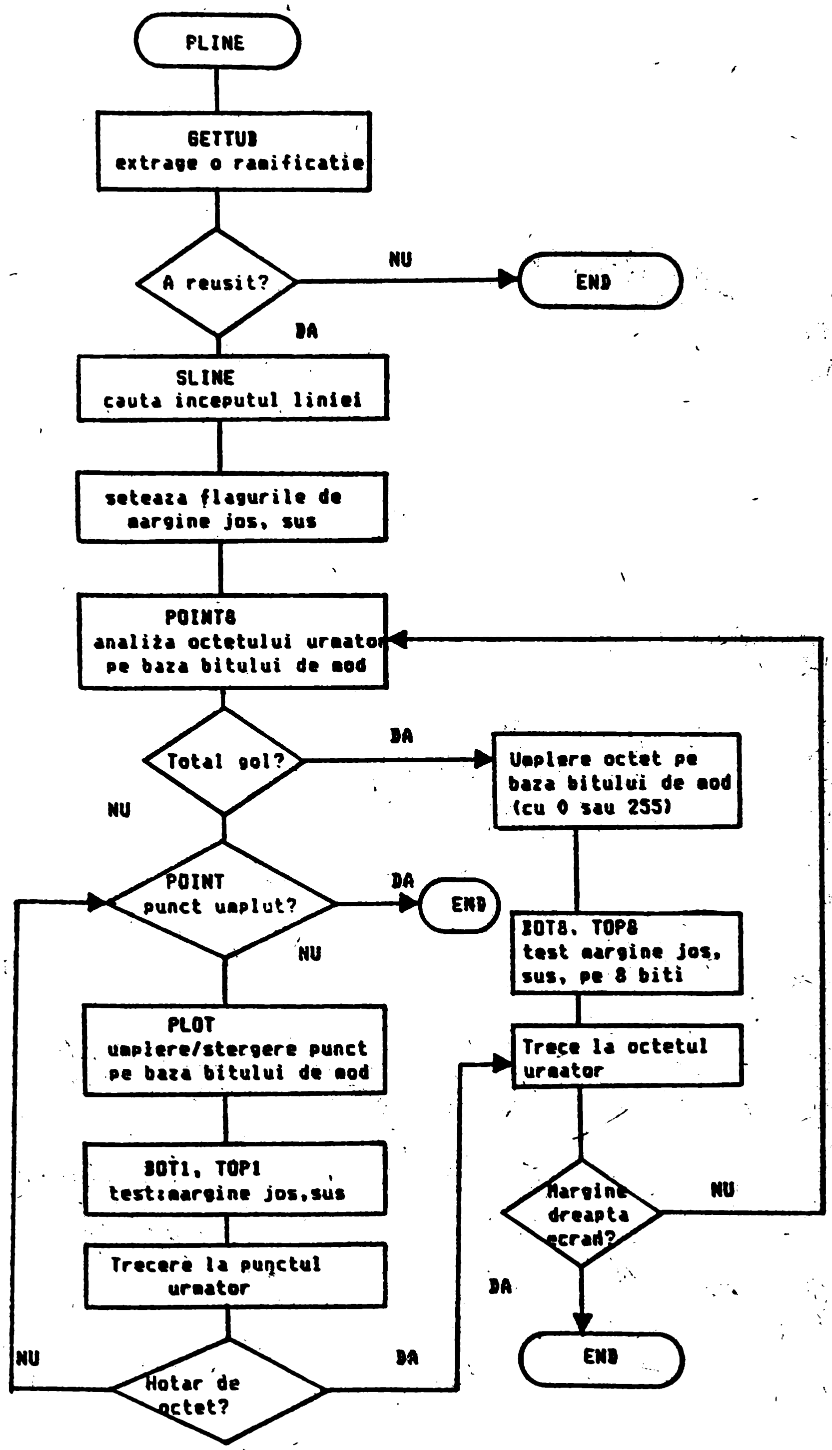
Următorul program BASIC exemplifică utilizarea rutinei (care se lansează de la adresa 63089), realizînd umplerea și apoi ștergerea unui cerc de rază 30. Utilizatorul va introduce coordonatele centrului cercului sau va modifica programul, astfel încît să se realizeze umplerea sau ștergerea pentru alte contururi.

```
 9 INPUT I
10 CIRCLE I,I,30
20 PLOT INVERSE 1;I+1
30 RANDOMIZE USR 6308'
32 PAUSE 0
40 PLOT I+1,I+1
50 RANDOMIZE USR 6308
60 GO TO 9
```

În vederea funcționării pe alte calculatoare care au memorie ecran astfel organizată sînt necesare mai multe modificări. De exemplu, pentru calculatorul PRAE 48 Ko (la care adresa de început a memoriei ecran este E000, iar cea de sfîrșit FFFF) sînt necesare următoarele modificări:

```
1000          Modificari      pentru
1001          memoria         ecran
ca la
1002          PRAE
1004
1006 SCREEN:
1008          EQU   #E000
1010
1012 SCREND:
1014          EQU   #FFFF
1016
1018 PIXADD:
1020          XOR   A
1022          SRA   B
1024          RR    C
1026          RRA
1028          SRA   B
1030          RR    C
1032          RRA
1034          SRA   B
1036          RR    C
1038          RLA
1040          RLA
1042          RLA
1044          LD    HL,SCREEN
1046          ADD   HL,BC
1048          RET
1050
1052 HLVP:
1054          PUSH  BC
1056          XOR   A
1058          LD    BC,32
1060          SBC   HL,BC
1062          POP   BC
1064          RET
1066
1068 HLDOWN:
1070          PUSH  BC
1072          LD    BC,32
1074          ADD   HL,BC
1076          POP   BC
1078          RET
```

Rutina a fost experimentată de Szabo Attila și Toth Levente (Liceul Bolyai, Tg. Mureș) la tabăra de informatică pentru elevii de liceu de la Tîrgu Mureș, 1989.



Schema logică pentru rutina PLINE

(realizează umplerea propriu-zisă)

Laborator Spectrum Sinclair


MACWORLD
VS
NETWORK WORLD
Desert Storm puts pressure on U.S. nets
Tools for
Satellites help U.S. forces control the skies over Iraq
COMPUTERWORLD
Blah Busters!
INFO WORLD
Commodore
RUN
Tips For PROGRAMMING SUCCESS
Publish
CD-ROMs
Toată informația
este pentru
Digital NEWS
DEC 2Q results, layoffs
Storage Concepts RAID
20/20 v 3.0 for VMS
DECUS hacker dispute
Storage must be managed
Token Ring an obstacle to full net access
IDG
INTERNATIONAL DATA
acoperă
Cockpit Maps Help Pilot Gulf Sorties
Publish
CIO
RE-ENGINEERING BUSINESS PROCESSES
PORTABLE NOTEBOOK COMPUTERS
PORTABLE OFFICE
Cellular Across America